

Avances en Inteligencia Artificial

Research in Computing Science

Series Editorial Board

Comité Editorial de la Serie

Editors-in-Chief:

Editores en Jefe

Grigori Sidorov (Mexico)
Gerhard Ritter (USA)
Jean Serra (France)
Ulises Cortés (Spain)

Associate Editors:

Editores Asociados

Jesús Angulo (France)
Jihad El-Sana (Israel)
Jesús Figueroa (Mexico)
Alexander Gelbukh (Russia)
Ioannis Kakadiaris (USA)
Serguei Levachkine (Russia)
Petros Maragos (Greece)
Julian Padget (UK)
Mateo Valero (Spain)

Editorial Coordination:

Coordinación Editorial

Socorro Méndez Lemus

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 62**, mayo 2013. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121511550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 62**, May 2013. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 62

Volumen 62

Avances en Inteligencia Artificial

Editores del volumen:

Volume Editors

Miguel González Mendoza
Félix Castro Espinoza

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2013



ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2013

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional “Adolfo López Mateos”, Zacatenco
07738, México D.F., México

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Prefacio

El propósito de este volumen es reflejar las nuevas direcciones de investigación en el área de Inteligencia Artificial. Los artículos de este volumen fueron seleccionados con base en un estricto proceso de revisión efectuada por los miembros del Comité editorial, tomando en cuenta la originalidad, aportación y calidad técnica de los mismos. Cada artículo fue revisado por tres miembros del comité editorial del volumen.

Este volumen contiene 42 artículos relacionados con varios aspectos de la Inteligencia Artificial. Los principales aspectos de los artículos corresponden a las áreas temáticas en las que más esfuerzo de investigación se realiza, es decir, aquellas que representan el estado del arte actual de la Inteligencia Artificial. Los artículos se dividen en las siguientes 6 áreas temáticas:

- Minería de datos (6 artículos),
- Procesamiento de lenguaje natural (7 artículos),
- Visión por computadora y procesamiento de imágenes (8 artículos),
- Redes neuronales y sistemas híbridos inteligentes (8 artículos),
- Lógica y ontologías (5 artículos),
- Robótica e interfaces inteligentes (8 artículos).

Este volumen de la revista puede ser interesante para los investigadores y estudiantes de las ciencias de la computación, especialmente en áreas relacionadas con la inteligencia artificial y su aplicación a los diferentes ámbitos de la vida cotidiana; así como, para el público en general interesado en este fascinante tema.

Expresamos nuestro agradecimiento por su valiosa ayuda en la preparación de este número especial de la revista RCS al Ing. Juan de Dios Noche Buena, Encargado de la Rectoría de la Universidad Politécnica de Francisco I. Madero (UPFIM), y a la Dirección de carrera de la Ingeniería en Sistemas Computacionales de la UPFIM, la cual impulsa nuevos horizontes de la ciencia y la tecnología a la comunidad universitaria del Valle del Mezquital en el Estado de Hidalgo, México.

El proceso de revisión y selección de artículos se llevó a cabo usando el sistema libremente disponible EasyChair, www.EasyChair.org. Para el diseño de la portada se usó la imagen <http://www.wallpaperpin.com/webdisk/fractal-dark-desktop-wallpaper.jpg>.

Mayo, 2013

Miguel González Mendoza
Félix Castro Espinoza

Índice

Pág.

Minería de datos

Metodología de minería de datos para perfilamiento cuantitativo de la brecha digital de ciudades	3
<i>Sergio R. Coria, Mónica Pérez-Meza, Rosibelda Mondragón-Becerra y Darío Barragán-López</i>	
Optimización de recurso para el tratamiento de grandes volúmenes de datos	15
<i>Mónica Santibáñez, Rosa María Valdovinos, Erendira Rendón, Roberto Alejo y J. Raymundo Marcial-Romero</i>	
Sistema de enseñanza para la técnica de clasificación de árboles de decisión.....	25
<i>Anilu Franco-Arcega, Félix Castro-Espinoza y Patricia Cortes-García</i>	
Planificación de dietas con Answer Set Programming para pacientes con Diabetes Mellitus 2	35
<i>Rosa Elena Pérez Ramírez, Claudia Zepeda Cortés e Irma del Carmen Zamora Ginez</i>	
Hacia la detección de impostores mediante un modelo basado en tareas	45
<i>Jorge Rodríguez-Ruiz, Raúl Monroy y J. Benito Camiña-Prado</i>	
WUIL: una base de datos para probar mecanismos de detección de intrusos.....	57
<i>J. Benito Camiña y Raúl Monroy</i>	

Procesamiento del lenguaje natural

Análisis de polaridad en Twitter.....	69
<i>Gilberto Leon Martagón, Esaú Villatoro-Tello, Héctor Jiménez-Salazar y Christian Sánchez-Sánchez</i>	
Aplicación de la implicación textual en un sistema de la validación de respuestas automáticas	79
<i>Partha Pakray, Alexander Gelbukh y Sivaji Bandyopadhyay</i>	
Desarrollo de una aplicación para el diálogo en lenguaje natural con un robot móvil.....	95
<i>Grigori Sidorov</i>	
Clasificación de servicios Web empleando estrategias supervisadas	109
<i>Víctor O. Díaz-Torres, Esaú Villatoro-Tello, Christian Sánchez-Sánchez y Héctor Jiménez-Salazar</i>	
Representación basada en grafos para la identificación de autoría en textos para el idioma español	119
<i>Nahun Loya, Iván Olmos, David Pinto y Jesús González</i>	

Clasificación de publicaciones en redes sociales semánticas mediante aprendizaje artificial con redes Bayesianas.....	129
<i>J. Carlos Conde-Ramírez, Pablo Camarillo-Ramírez y Abraham Sánchez-López</i>	
Clasificación de acosadores en línea utilizando Bootstrapping	139
<i>Yuridiana Alemán, Darnes Vilariño y David Pinto</i>	

Visión por computadora y procesamiento de imágenes

Segmentación de objetos en movimiento por flujo óptico y color sin información a priori de la escena	151
<i>Angel Juan Sánchez García y Homero Vladimir Ríos Figueroa</i>	
Recuperador de imágenes en base a contenido: una aplicación con esporas	161
<i>Nayeli Joaquinita Meléndez Acosta, Homero Vladimir Ríos Figueroa y Antonio Marín Hernández</i>	
Segmentación de imágenes para el análisis de lesiones en la piel: un estudio.....	171
<i>Juan García y Salvador Venegas</i>	
Autenticación de documentos por medio de la extracción de características de las fibras del papel utilizando Wavelets.....	181
<i>Rodolfo Ibarra-Orozco, Juan Carlos López-Pimentel, Miguel González-Mendoza y Víctor Ramos-Fon Bon</i>	
Método robusto para detectar dedos usando profundidad	189
<i>Isaac Juan Rudomín Goldberg, Jorge Adolfo Ramírez Uresti y Christian Jesús Arzate Cruz</i>	
Análisis de componentes principales aplicado al proceso de reconocimiento de objetos 3D bajo rotación.....	197
<i>Manuel Ornelas, Valentín Calzada, Hector Puga, Raúl Santiago y Martín Carpio</i>	
Algoritmo de segmentación para imágenes reales para la clasificación de cetáceos	209
<i>David Trejo-Salazar, Blanca Esther Carvajal-Gámez y Francisco Javier Gallegos-Funes</i>	
Segmentación de peatones a partir de vistas aéreas.....	219
<i>Jorge Ivan Rivalcoba Rivas e Isaac Rudomín</i>	

Redes neuronales y sistemas híbridos inteligentes

Clasificación de arritmias cardiacas utilizando knn y Naive Bayes mejorada con algoritmos genéticos (AG) y optimización de cúmulo de partículas (PSO).....	233
<i>Christian Padilla-Navarro, Rosario Baltazar-Flores, David Cuesta-Frau y Victor Zamudio-Rodríguez</i>	

Comparativa entre los algoritmos DE y PSO aplicada a la demodulación de interferogramas de franjas cerradas modeladas por polinomios de Zernike	243
<i>Juan de Anda, Martín Carpio, Héctor J. Puga, Luis Mancilla Espinoza, Manuel Ornelas, Rosario Baltazar y Marco Sotelo</i>	
Predicción de índices de reprobación escolar: un modelo de series de tiempo	253
<i>Rosa-Irene Rojas-Rauda y Omar-Jacobo Santos-Sánchez</i>	
Modelado de sistemas adaptativos de inferencia neuro-difusa usando diseño central compuesto	259
<i>Pascual Noradino Montes Dorantes, Rolando Praga Alejo, Juan Pablo Nieto González y Gerardo Maximiliano Méndez</i>	
Sistema de predicción de patrones de juego	271
<i>Aram B. González y Jorge A. Ramírez Uresti</i>	
Análisis del desempeño de los algoritmos redes neuronales de impulso (SNN) y máquinas de soporte vectorial (SVM) en programación paralela	281
<i>Israel Tabarez Paz, Neil Hernández Gress y Miguel González Mendoza</i>	
Estudio comparativo de estrategias de control no lineal basadas en pasividad para sistemas robóticos	295
<i>Miguel O. Uribe H., José M. Fernández R., Juan D. Ramírez Z., Luis R. Lechuga G., Omar A. Domínguez R. y Luis E. Ramos V.</i>	
Sintonización y optimización de un PID con algoritmos evolutivos	305
<i>Armando Pedraza Martín, Omar Santos y Félix Castro</i>	

Lógica y ontologías

Comportamientos en simulación de multitudes: revisión del estado del arte	319
<i>Oriam De Gyves, Leonel Toledo e Isaac Rudomín</i>	
Hacia la formación automática de ontologías en documentos no estructurados	335
<i>Martha Idalid Rivera González, Adolfo Guzmán Arenas y Miguel González Mendoza</i>	
Descripción de un proceso de enriquecimiento de ontologías obtenidas de base de datos relacionales	343
<i>Joanna Alvarado-Uribe, David González-Marrón, Miguel González-Mendoza y Angélica Enciso-González</i>	
Análisis y evaluación automatizada de modelos paramétricos CAD	361
<i>Daniel García Calderón, Nestor Velasco Bermeo y Miguel González Mendoza</i>	
Distribución balanceada de ayuda humanitaria resuelto con un algoritmo metaheurístico	373
<i>Ivo Erasmo Buzón Cantera, Jaime Mora Vargas, Ángel Ruiz Bartolomé y Patrick Soriano</i>	

Robótica e interfaces inteligentes

Control sub-óptimo con desempeño robusto para regulación global de robots manipuladores	393
<i>Mario O. Ordaz-Oliver, Omar J. Santos-Sánchez y Omar A. Domínguez-Ramírez</i>	
Modelo teórico y generación de instancias de prueba para problemas RCCN (Reducing Collisions in Constrained Navigation)	405
<i>Josué Oregel, Héctor J. Puga, Martín Carpio, Víctor Zamudio, Manuel Ornelas y Luis Mancilla Espinoza</i>	
Captura de movimiento de brazo derecho mediante una plataforma de evaluación para tener un banco de datos: Estudio experimental en paciente hemipléjico	417
<i>Abraham Briseño Cerón, Omar A. Domínguez Ramírez, Vicente Parra Vega y Dora A. Rodríguez Vega</i>	
Pruebas de laberintos 2D y 3D con propósitos de uso en rehabilitación neuropsicológica	427
<i>Víctor Tomás Tomás Mariano, Felipe de Jesús Núñez Cárdenas, Raúl Hernández Palacios y Mariano Pozas Cárdenas</i>	
Desarrollo de un lápiz óptico inalámbrico para aplicaciones de rehabilitación motriz fina	437
<i>Roberto A. Montaña-Murillo, Rubén Posada-Gómez, Albino Martínez-Sibaja, Patricia I. Cornelio-Martínez, Blanca E. González-Sánchez y Alberto A. Aguilar-Lasserre</i>	
Diseño de un sistema auxiliar para rehabilitación motriz mediante realidad virtual y valoración de la fuerza palmar en deficiencias de motricidad fina	447
<i>Patricia Ivette Cornelio Martínez, Blanca Estela González Sánchez, Roberto Antonio Montaña Murillo, Rubén Posada Gómez y Gerardo Águila Rodríguez</i>	
Construcción de un sistema de teclado virtual como apoyo pedagógico para personas con habilidades motrices limitadas	459
<i>Eduardo Emmanuel Ochoa Gracia, José Federico Ramírez Cruz, Miquelina Sánchez Pulido y Blanca E. Pedroza Méndez</i>	
Guiado háptico pasivo sub-óptimo con observador de energía	469
<i>Mario O. Ordaz-Oliver, Omar A. Domínguez-Ramírez y Omar J. Santos-Sánchez</i>	
Índice de autores	483
Comité editorial del volumen	485

Minería de datos

Metodología de minería de datos para perfilamiento cuantitativo de la brecha digital de ciudades

Sergio R. Coria, Mónica Pérez-Meza, Rosibelda Mondragón-Becerra
y Darío Barragán-López

Instituto de Informática
Universidad de la Sierra Sur
Calle Guillermo Rojas Mijangos S/N, Esq. Av. Universidad Col. Ciudad Universitaria
Míahuatlán de Porfirio Díaz, Oax.,
México
{coria, mperez, rmondragon, dbarragan}@unsis.edu.mx
<http://www.unsis.edu.mx>

Resumen Se propone un enfoque novedoso para análisis y modelación del fenómeno de la brecha digital, concentrándose en el nivel de ciudad como unidad territorial. Se usan los algoritmos PART and J4.8 de Witten y Frank, implementados en el reconocido toolkit WEKA, sobre datos que describen diversos niveles de brecha digital en ciudades. Se implementan datasets del Censo 2010 de Población y Vivienda de México. Se selecciona la mayoría de sus variables y se transforman en porcentajes o en promedios por municipio. Se discretiza el porcentaje de hogares que tienen Internet para usarse como *target*. La mayoría de las otras variables transformadas se introducen como predictores a los algoritmos. Los resultados muestran que nuestro enfoque es altamente útil para producir perfiles de ciudades que describen interacciones entre variables demográficas, socio-económicas y educacionales asociadas a diversos niveles de presencia de Internet. Finalmente, se discuten las ventajas de estos algoritmos para este dominio.

Palabras clave: algoritmo PART, algoritmo J4.8, datos de censos, brecha digital, exclusión digital, inclusión digital, WEKA.

1. Introducción

La brecha digital es definida como *la distancia entre individuos, hogares, negocios y áreas geográficas en diferentes niveles socio-económicos respecto tanto a sus oportunidades para acceder a tecnologías de información y comunicación (TIC) como para su uso del Internet para una amplia variedad de actividades* [1]. Diversas teorías sobre el fenómeno de la brecha digital consideran que esta puede ser explicada por la interacción entre la presencia de bienes y servicios de TIC y un conjunto de características demográficas, socio-económicas y educacionales de la población en el territorio que sea estudiado.

Uno de los principales propósitos de la investigación de la brecha digital, tanto de países como de regiones o de ciudades, es descubrir patrones que ofrezcan conocimiento acerca de las interacciones entre la presencia de bienes y servicios de TIC en hogares y las características demográficas, socio-económicas y educacionales de sus habitantes. La investigación de la brecha digital requiere técnicas que faciliten el descubrimiento de relaciones causales entre variables. Por lo tanto, las técnicas de clasificación automática [2] pueden ser de gran utilidad.

La presencia de Internet en hogares es uno de los principales aspectos de las TIC involucrados en el fenómeno. Por ello, el problema de investigación específico en este trabajo consiste en evaluar cómo los algoritmos de clasificación automática PART [3] y J4.8 [4] (basado en C4.5 [5]) pueden contribuir a descubrir conocimiento en el campo de la investigación de la brecha digital, enfocándose en el nivel de ciudad como unidad para análisis y modelación y teniendo como *target* la presencia de Internet en hogares. El propósito principal no es explotar los modelos producidos para automatizar tareas de clasificación, sino ofrecer a los investigadores de la brecha digital una técnica que produzca modelos descriptivos capaces de guiar el descubrimiento de explicaciones para este fenómeno en ciudades de países específicos.

En esta investigación se eligió a J4.8 y PART porque, desde una perspectiva teórica, ambos forman parte de la familia de algoritmos de aprendizaje automático supervisado, lo cual permite clasificar instancias, descubriendo los patrones que caracterizan a sus clases. También, porque la precisión de ambos algoritmos es comparable a la de otros más complejos como el perceptrón multicapa (*multi-layer perceptron*, MLP). Desde una perspectiva práctica, fueron elegidos porque ofrecen una mayor facilidad explicativa y de interpretación a usuarios de dominio que no tienen conocimientos de aprendizaje automático y además porque facilitan la programación de sistemas de apoyo a la toma de decisiones (*decision support systems*, DSS).

El artículo está organizado en las siguientes secciones: la sección 2 presenta un breve panorama del estado del arte. La sección 3 describe los datos fuente que se usan para desarrollar la metodología y para realizar su evaluación. La sección 4 explica la metodología propuesta por esta investigación. La sección 5 presenta los resultados, que están organizados como un conjunto de siete modelos PART y siete modelos J4.8. La sección 6 discute los resultados. Finalmente, la sección 7 presenta algunas conclusiones y sugiere trabajo futuro en este campo.

2. Estado del arte

Uno de los principales propósitos de la investigación del fenómeno de la brecha digital consiste en identificar las variables que lo determinan en contextos territoriales determinados. En la mayoría de los trabajos relacionados con el tema, la unidad territorial más frecuente en análisis y modelación es el nivel de país. Por ejemplo, en [6] se analiza las determinantes de la brecha digital global realizando un análisis entre países acerca de la penetración de computadora e Internet.

Las técnicas más comunes para modelar y analizar la brecha digital aplican índices compuestos. Por ejemplo, [7] presenta el Índice de Acceso Digital (*Digital Access Index*, DAI), que es uno de los más utilizados actualmente. [8] propone un índice para análisis de infraestructura y acceso de TIC en países. Otro enfoque frecuente es el de análisis multivariable. Por ejemplo, [9] realiza un análisis crítico de una serie de índices de medición de brecha digital y propone como alternativa el enfoque multivariable y [10] hace un análisis empírico multivariable de brecha digital entre países. En [11] se describe una serie de retos metodológicos acerca de la medición de la brecha digital; entre estos, las deficiencias o errores frecuentes en el uso de índices compuestos, de los análisis multivariable y de los que incluyen la variable tiempo.

La ciudad como unidad territorial de análisis y modelación se ha vuelto relevante en la investigación de la brecha digital porque los gobiernos y las empresas definen políticas y estrategias dirigidas a ese ámbito. Muchos de los datos requeridos para ello están disponibles en censos nacionales. Al respecto, se han aplicado técnicas de minería de datos sobre censos, aunque en otras áreas distintas del fenómeno de la brecha digital. Algunos trabajos relacionados son, por ejemplo: [12], que hace un estudio de las clases sociales marginadas de Taiwán, y [13], que aplica el algoritmo CART [14] sobre datos del censo de China.

Pocos trabajos relacionados aplican aprendizaje automático al estudio de la brecha digital. Por ejemplo, [15] realiza una medición de la brecha digital internacional aplicando mapas auto-organizantes de Kohonen. Un trabajo similar a la presente investigación es [16], en el cual se muestran resultados sobre el uso de árboles clasificadores J4.8 para análisis y modelación de brecha digital de municipios mexicanos. Sin embargo, en ese trabajo no se usa el algoritmo PART, cuyo desempeño en diversos dominios es mejor que J4.8, según evaluaciones previas realizadas por [3].

J4.8 está inspirado en el reconocido algoritmo C4.5 [5]. A su vez, PART está basado en J4.8; por ello, en esta investigación se espera que algunos de los resultados producidos por PART sean similares a los producidos por J4.8. También, se espera que los modelos PART tengan un número de reglas menor que las producidas por J4.8 y que los valores de *accuracy* de PART sean mayores o iguales que los de J4.8.

3. Datos fuente

Los modelos se crean usando *datasets* experimentales producidos con la base de datos del Censo 2010 de Población y Vivienda de México [17]. Su diccionario de datos [18] es de gran utilidad para estos propósitos. La mayoría de sus ítems son frecuencias absolutas de variables demográficas, socio-económicas y educacionales de los habitantes o de las viviendas. Ofrece 200 atributos, organizados en 14 categorías, de 2,456 municipios. Las categorías, con sus respectivos números de atributos, son: identificación geográfica (9), población (47), fecundidad (1), migración (12), población indígena (13), discapacidad (9), educación (42), características económicas (12), servicios de salud (6), situación conyugal (3), religión (4), hogares censales (6), características de vivienda (35) y tamaño de localidad (1).

Una pequeña cantidad de estos atributos no son frecuencias absolutas, sino promedios; por ejemplo, de: hijos nacidos vivos, ocupantes por casa, años de educación, etc. Esta diferenciación entre frecuencias absolutas y promedios es relevante porque cada uno de estos dos tipos de atributos es procesado en forma distinta para propósitos de representación de la información. Una vez seleccionados los atributos, se crea una serie de *datasets* experimentales. No es solamente un *dataset* porque se usan diferentes tamaños de intervalo que definen diferentes conjuntos de etiquetas de clase para el atributo *target*.

4. Método

Para crear los modelos se realiza un proceso típico de minería de datos con base en los siguientes pasos generales: 1) selección de datos, 2) cálculo y discretización, 3) organización de *datasets*, y 4) creación y evaluación de modelos. Un aspecto destacable en este proceso es la discretización del porcentaje de hogares que tienen Internet en cada municipio, el cual se transforma en una etiqueta nominal que se usa como *target* en los modelos. A continuación, se describe cada paso del proceso.

4.1. Selección de datos

La gran mayoría de las variables del censo son seleccionadas para crear los *datasets* experimentales y solamente se descarta un pequeño conjunto de aquellas. Tres de los nueve atributos geográficos disponibles en la fuente se incorporan en los *datasets* experimentales: *LONGITUD*, *LATITUD* y *ALTITUD*. Los seis atributos descartados son dos identificadores de localidad, dos de municipio y dos de entidad federativa que no contribuyen al descubrimiento de patrones útiles en este caso.

Aunque algunos atributos están correlacionados estadísticamente en la fuente original (p. ej. población *total* entre 0 y 2 años de edad, con población *masculina* entre 0 y 2 años de edad, y población *femenina* entre 0 y 2 años de edad), no se realiza una reducción manual de características basada en análisis de correlación antes de generar los modelos. La razón es que PART y J4.8 omiten automáticamente de los modelos a los atributos que tienen una baja capacidad discriminativa y preservan a los que tienen mayor capacidad. Así, se adopta un enfoque exploratorio para descripción de los perfiles de ciudades considerando diversos niveles de presencia de Internet en hogares.

4.2. Cálculo y discretización de datos

De los atributos seleccionados, se calculan porcentajes por municipio a partir de las frecuencias absolutas, ya sea respecto al número total de hogares o al de habitantes, según corresponda a cada variable. Los atributos que son promedios en los datos fuente (p. ej. *GRAPROES*, promedio de años de educación de los habitantes del municipio) son incorporados en los *datasets* experimentales sin realizar ninguna transformación. El propósito es usar valores normalizados, ya sea porcentajes o

promedios. La excepción es un pequeño número de atributos que se usan como frecuencias absolutas o valores adimensionales; p. ej. las tres coordenadas geográficas. El atributo *POBTOT*, número total de habitantes de municipio, se incorpora directamente en los datasets sin aplicarle transformaciones aritméticas porque se supone que puede ser útil para distinguir perfiles de municipios respecto a la presencia de Internet.

Se realiza un proceso de discretización solamente al porcentaje de presencia de Internet en hogares (*VPH_INTER_%*). El propósito es producir un nuevo atributo, *VPH_INTER_presence*, de tipo nominal, usándolo como *target* para crear modelos PART y J4.8. Las interacciones entre este atributo y todos los demás tienen potencial para describir una parte considerable del fenómeno de la brecha digital.

La discretización de *VPH_INTER_%* se basa en un conjunto de siete tamaños de intervalo que se usan para crear siete datasets. La motivación para considerar diferentes tamaños de intervalo es evaluar su impacto en: 1) el número de clases mayoritarias generadas y sus porcentajes en cada dataset, y 2) accuracy de los modelos. El número de siete tamaños de intervalo es convencional; estos son: 2, 4, 5, 6, 10, 20 y 25 puntos porcentuales. Así, los valores de *VPH_INTER_%* son transformados en las etiquetas de clase de *VPH_INTER_presence*: $c_1, c_2, c_3, \dots, c_n$. La Tabla 1 presenta los datasets experimentales con sus respectivos tamaños de intervalo, número de clases y porcentajes de clases mayoritarias.

Table 1. Siete datasets experimentales que contienen a *VPH_INTER_presence* como *target* para modelos.

Dataset No.	Tamaño de intervalo (puntos porcentuales)	No. de clases	Clases mayoritarias
1	2	100/2=50	c_{50} (37.8%) + c_{49} (15.3%) + c_{48} (10.5%) + c_{47} (8.0%) + c_{46} (6.1%) + c_{45} (4.2%) = 82.0%
2	4	100/4=25	c_{25} (53.1%) + c_{24} (18.5%) + c_{23} (10.4%) = 82.0%
3	5	100/5=20	c_{20} (59.1%) + c_{19} (18.6%) + c_{18} (9.6%) = 87.3%
4	6 (de la fórmula de Sturges [18])	68.2/6=11.3	c_{12} (63.6%) + c_{11} (18.4%) = 82.0%
5	10	100/10=10	c_{10} (77.7%) + c_9 (14.5%) = 92.2%
6	20	100/20=5	c_5 (92.2%)
7	25	100/25=4	c_4 (95.0%)

El tamaño de intervalo 6 es especial porque está determinado con una técnica [19] para calcular tamaño óptimo dependiendo del rango y del número de instancias de la variable en el dataset: $C = \text{rango} / (1 + 3.322 \log N)$, donde C es el tamaño óptimo, $\text{rango} = \text{máximo} - \text{mínimo}$ y N es el número de instancias. En los otros tamaños de

intervalo, la escala es de 0 a 100%; en cambio en este, es del valor mínimo (0.0%) al máximo (68.2%) de la variable $VPH_INTER_%$. Una vez generados los valores nominales de $VPH_INTER_presence$, el atributo $VPH_INTER_%$ es eliminado de los datasets para evitar la generación de un patrón trivial que vincule a este valor porcentual con el nominal. La asignación de etiquetas de clase al atributo $target\ VPH_INTER_presence$ está basada en las siguientes reglas: 1) la mejor clase es c_1 y corresponde a los porcentajes más altos de $VPH_INTER_%$; 2) la peor clase es c_n , donde n es igual al número de clases del *dataset* y corresponde a los porcentajes más bajos de la variable.

4.3. Organización de los *datasets*

Se producen siete *datasets* para el $target\ VPH_INTER_presence$. En cada uno, el número de atributos es 183 (incluyendo al $target$). La mayoría de los predictores son porcentajes que describen características de la población, de la vivienda y de los hogares. Un pequeño número de predictores son promedios municipales de años de educación, de hijos nacidos vivos, de ocupantes por casa, entre otros, así como valores absolutos de población total y de las tres coordenadas geográficas. Los análisis de Pareto de las clases muestran las mayoritarias y la existencia de un desbalance entre las clases de cada *dataset*: la mayoría de las instancias corresponde a las clases con menor presencia de Internet. Esto podría ser relevante porque el desbalance podría sesgar a los modelos, haciéndolos más capaces de reconocer a las clases mayoritarias. Sin embargo, como el principal propósito de estos modelos es descriptivo y no para clasificación automática, esta situación no es tan relevante.

4.4. Creación y evaluación de modelos

Los modelos PART y J4.8 en esta investigación se crean usando algoritmos implementados en el *toolkit* WEKA [4] con sus parámetros de configuración por omisión: para PART, -M 2 -C 0.25 -Q 1; para J4.8, -C 0.25 -M 2. En ambos tipos de modelos, el modo de prueba seleccionado es Validación Cruzada de 10 Subconjuntos (*10-fold cross validation*). Para cada *dataset*, se crea un modelo PART y un modelo J4.8. Los criterios de aceptación en la etapa de evaluación son: 1) el *accuracy* debe ser mayor o igual que 75%; 2) el estadístico Kappa [20] debe ser mayor o igual que 0.67; 3) el número de clases mayoritarias con base en análisis de Pareto en el correspondiente *dataset* experimental debe ser mayor que 1. Las razones para estos criterios son: los umbrales de aceptación de *accuracy* y Kappa están definidos de modo convencional con base en valores frecuentemente usados en la literatura. El valor de umbral para el número de clases mayoritarias está establecido considerando que una gran mayoría de los municipios mexicanos pertenece a la peor clase en cada *dataset* y, por lo tanto, necesita aplicarse una restricción al desbalance de las clases.

4.5. Creación de perfiles cuantitativos por clase de municipio

Considerando que el *target* representa clases de municipios con distintos porcentajes de presencia de Internet en hogares, al agrupar las reglas de un modelo PART (o las hojas de un árbol J4.8) con base en sus clases, se está describiendo el perfil de cada clase de municipio. De este modo, el perfil p de una clase específica c_i está constituido por la suma lógica de sus reglas: $p(c_i) = r_1(c_i)$ or $r_2(c_i)$ or $r_3(c_i)$... or $r_n(c_i)$.

5. Resultados

La Tabla 2 resume los resultados de los modelos generados con los siete *datasets*, presentando sus *accuracies*, Kappas y número de reglas y de hojas (los modelos completos pueden solicitarse a los autores). Los modelos PART y J4.8 que cumplen mejor los criterios de aceptación corresponden al *dataset* No. 4. Estos modelos son identificados como P_4 y J_4 , respectivamente. De ellos, es preferible P_4 porque, aunque los valores de *accuracy* y de Kappa son muy similares a los de J_4 , el número de reglas de P_4 (67) es considerablemente menor que el de hojas de J_4 (120). Esta diferencia implica que P_4 puede describir el fenómeno en forma más compacta que J_4 .

Table 2. Resultados de modelos PART y J4.8 sobre siete datasets para el *target* *VPH_INTER_presence*.

Dataset No.	Modelos PART			Modelos J4.8				
	ID de modelo	Accu-racy %	Kappa	No. de reglas	ID de modelo	Accu-racy	Kappa	No. de hojas
1	P_1	53.1	0.4190	217	J_1	53.2	0.4199	345
2	P_2	70.9	0.5632	110	J_2	73.1	0.5691	205
3	P_3	78.9	0.6513	86	J_3	79.8	0.6645	150
4	P_4	82.2	0.6764	67	J_4	82.4	0.6797	120
5	P_5	91.7	0.7778	28	J_5	91.2	0.7639	67
6	P_6	98.2	0.8733	10	J_6	97.8	0.8428	17
7	P_7	99.2	0.9152	8	J_7	98.8	0.8786	9

Del modelo P_4 pueden obtenerse los perfiles de las ciudades (municipios), que se muestran a continuación:

$p(c_1) = r_{50}$ (la clase tiene solamente una regla).

$p(c_2) =$ (la clase no existe en el *dataset* y, por lo tanto, tampoco en los modelos).

$p(c_3) =$ (la clase tiene solo una instancia, por lo cual PART no la modela).

$p(c_4) = r_{67}$ (la clase tiene solamente una regla).

$p(c_5) = r_{60}$ (la clase tiene solamente una regla).

$p(c_6) = r_{59}$ or r_{63} .

$p(c_7) = r_{57}$ or r_{62} or r_{64} .

$p(c_8) = r_{53}$ or r_{61} or r_{66} .

$p(c_9) = r_{30}$ or r_{51} or r_{54} or r_{56} .

$p(c_{10}) = r_{14} \text{ OR } r_{23} \text{ OR } r_{31} \text{ OR } r_{36} \text{ OR } r_{41} \text{ OR } r_{43} \text{ OR } r_{48} \text{ OR } r_{52} \text{ OR } r_{55} \text{ OR } r_{58}.$

$p(c_{11}) = r_7 \text{ OR } r_{12} \text{ OR } r_{18} \text{ OR } r_{19} \text{ OR } r_{21} \text{ OR } r_{22} \text{ OR } r_{26} \text{ OR } r_{27} \text{ OR } r_{28} \text{ OR } r_{29} \text{ OR } r_{32} \text{ OR } r_{35} \text{ OR } r_{39} \text{ OR } r_{40} \text{ OR } r_{42} \text{ OR } r_{44} \text{ OR } r_{45} \text{ OR } r_{46} \text{ OR } r_{65}.$

$p(c_{12}) = r_1 \text{ OR } r_2 \text{ OR } r_3 \text{ OR } r_4 \text{ OR } r_5 \text{ OR } r_6 \text{ OR } r_8 \text{ OR } r_9 \text{ OR } r_{10} \text{ OR } r_{11} \text{ OR } r_{13} \text{ OR } r_{15} \text{ OR } r_{16} \text{ OR } r_{17} \text{ OR } r_{20} \text{ OR } r_{24} \text{ OR } r_{25} \text{ OR } r_{33} \text{ OR } r_{34} \text{ OR } r_{37} \text{ OR } r_{38} \text{ OR } r_{47} \text{ OR } r_{49}.$

Como ejemplo de las reglas, se comentan tres de estas a continuación:

r_{50} de c_1 : $P_3A5_M_porcent \leq 1.9 \text{ AND } P3A5_NOA_M_porcent \leq 0.5 \text{ AND } ALTITUD \leq 2249$: c_1 (2).

Significa: Si la población entre 3 y 5 años de edad de sexo masculino es menor o igual al 1.9% y la población de 3 a 5 años de sexo masculino que no asiste a la escuela es menor o igual 0.5% y la altitud geográfica del municipio es menor o igual a 2,249 metros, entonces el municipio es de clase 1, teniendo entre 66% y 72% de presencia de Internet en hogares (ocurre en 2 de los municipios, sin excepción).

r_1 de c_{12} : $VPH_PC_porcent \leq 11.9 \text{ AND } VPH_PC_porcent \leq 8.5$: c_{12} (1154).

Significa: si los hogares con PC son menores o iguales al 11.9% y además son menores al 8.5%, entonces el municipio es de clase 12, teniendo entre 0% y 6% de presencia de Internet en hogares (ocurre en 1,154 de los municipios, sin excepción).

r_{46} de c_{11} : $VPH_PC_porcent \leq 31.6 \text{ AND } PNACOE_porcent \leq 1.9$: c_{11} (5/1).

Significa: Si los hogares con PC son menores o iguales a 31.6% y la población nacida en otra entidad es menor o igual a 1.9%, entonces el municipio pertenece a la clase 11, teniendo entre 6% y 12% de presencia de Internet en hogares (ocurre en 5 de los municipios, excepto en uno que tiene las condiciones descritas pero no pertenece a la clase 11).

En cada perfil, son más significativas las reglas que tienen los mayores valores de soporte (*support*, *coverage*) y de confianza (*confidence*). El soporte se calcula como: $(a/n)*100$ donde a es el número de instancias que tienen las condiciones descritas por la regla (pertenecan, o no, a su clase) y n es el número total de instancias en el dataset (2,456). La confianza se calcula: $((a-b)/a)*100$, donde b es el número de instancias que no pertenecen a esa clase, es decir, las excepciones de la regla. Por ello, las reglas con mayor soporte describen a un mayor número de municipios y representan a los patrones más sólidos del fenómeno de la brecha digital en el país.

6. Discusión

El uso de la variable $VPH_INTER_presence$ como *target* se justifica porque representa a las diversas clases de municipios y es el principal medio para descubrir los patrones socio-demográficos de cada clase. Los predictores útiles para describir los patrones son descubiertos automáticamente por los algoritmos evaluados. La Tabla 2 muestra que las *accuracies* y Kappas son altamente similares entre cada par de modelos PART y J4.8 para cada *dataset*. La diferencia más significativa entre cada

par es el número de reglas y de hojas, respectivamente. Las hojas en los árboles J4.8 constituyen los elementos terminales, es decir, los *consecuentes*, de reglas que son comparables a las reglas de los modelos PART. En ambos modelos, las reglas representan conocimiento descubierto no trivial. La medida de soporte de cada regla representa qué tan frecuente es ese patrón en el *dataset*. Por ello, al identificar las reglas con mayor soporte en P_4 se obtienen los patrones más significativos del fenómeno.

La regla con mayor soporte ($1,154 / 2,456 * 100 = 47.0\%$) en P_4 es r_j ; por ello, puede afirmarse que el patrón más significativo en el análisis de la brecha digital en municipios de México consiste en que: *los municipios con menor presencia de Internet (de 0 a 6% de los hogares) se caracterizan principalmente porque la presencia de PC es menor al 8.5%* (esta regla ha sido simplificada eliminando el valor 11.9%).

Contrastando los resultados con los de trabajos relacionados, la mayoría de estos últimos no son comparables porque aplican técnicas distintas para análisis y modelación y porque su unidad de análisis es el nivel de país, no el de ciudad. El único trabajo comparable es [15] porque en él se usan árboles clasificadores J4.8 con datos de municipios. Al comparar los dos trabajos, se observa que J4.8 y PART resultan útiles para este dominio porque ofrecen modelos descriptivos capaces de guiar el descubrimiento de explicaciones causales. Sin embargo, también se confirma que PART ofrece valores de *accuracy* y Kappa que son similares a los de J4.8, pero con PART los modelos tienen menos reglas y, en algunos casos, estas son más compactas. En ambos algoritmos, los perfiles de clases de municipios con distintos grados de presencia de Internet pueden generarse agrupando las reglas que corresponden a cada clase.

7. Conclusiones y trabajo futuro

Este artículo ha propuesto una metodología basada en la aplicación de los algoritmos PART y J4.8 para investigar la brecha digital de ciudades (particularmente, *municipios*) describiendo interacciones entre la presencia de Internet en hogares y una serie de aspectos demográficos, socio-económicos y educacionales de los habitantes. Con base en resultados empíricos, nuestra postura es que tanto PART como J4.8 son útiles para el análisis y la modelación del fenómeno de la brecha digital de ciudades porque: 1) las *accuracies* y Kappas de los modelos son satisfactorias sobre la base de criterios generalmente aceptados en el campo del aprendizaje automático, y 2) se descubren variables clave para la descripción y potencial explicación del fenómeno. Sin embargo, como era esperado con base en trabajos relacionados acerca de estos dos algoritmos, las reglas producidas por PART son menos y más cortas que las producidas por J4.8.

Nuestras principales contribuciones científicas en esta investigación son: 1) la propuesta de una técnica novedosa basada en PART y J4.8 para investigar el fenómeno de la brecha digital de ciudades, aplicable sobre datos de censos nacionales

de diversos países, y 2) el descubrimiento de patrones no triviales de este fenómeno en el contexto específico de los municipios de México.

En trabajo futuro se debería generar y evaluar otros modelos PART y J4.8 usando como *targets* otros datos respecto a presencia de bienes y servicios de TIC en hogares, tales como PC, teléfono fijo o teléfono celular. Los atributos de entidad federativa podrían incluirse dentro de los predictores. También, podría realizarse un análisis comparativo (*benchmarking*) entre estos dos algoritmos y uno o dos más usando los mismos datasets. Finalmente, deberían crearse otros modelos usando datos de censos de otros países.

Referencias

1. Organisation for Economic Co-operation and Development (OECD). Glossary of Statistical Terms. <http://stats.oecd.org/glossary/index.htm>. Accesado 01/Abr/2013.
2. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques (3rd ed.). The Morgan Kaufmann Series in Data Management Systems, Waltham (2011).
3. Frank, E., Witten, I.H.: Generating Accurate Rule Sets Without Global Optimization. In: 15th IMLS International Conference on Machine Learning, pp. 144–151. Morgan Kaufmann Publishers Inc., San Francisco (1998).
4. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques (3rd ed.). Morgan Kaufmann Series in Data Management Systems, Burlington (2011).
5. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993).
6. Chinn, D.M., Fairlie, W.R. The Determinants of the Global Digital Divide: A Cross-country Analysis of Computer and Internet Penetration. *Oxford Economic Papers* 59(1), 16–44 (2007).
7. International Telecommunications Union (ITU). World Telecommunication Development Report. Access Indicators for the Information Society. Accesado 13/Ene/2013. http://www.itu.int/ITU-D/ict/publications/wtdr_03/material/DAI.pdf
8. Hanafizadeh, M.R., Saghaei, A., Hanafizadeh, P. An Index for Cross-country Analysis of ICT Infrastructure and Access. *Telecommunications Policy* 33, 385–405 (2009).
9. Bruno, G., Esposito, E., Genovese, A., Gwebu, K.L. A Critical Analysis of Current Indexes for Digital Divide Measurement. *The Information Society* 27, 16–28 (2011).
10. Iliadis, M.S., Paravantis, J.A. A Multivariate Cross-country Empirical Analysis of the Digital Divide. In: *Procs. of ISCC '11 2011 IEEE Symposium on Computers and Communications*, pp. 785–788. Washington, D.C., U.S.A. (2011).
11. Vehovar, V., Sicherl, P., Husing, T., Dolnicar, V.: Methodological Challenges of Digital Divide Measurements. *The Information Society* 22, 279–290 (2006).
12. Chang, C.J., Shyue, S.W. A Study on the Application of Data Mining to Disadvantaged Social Classes in Taiwan Population Census. *Expert Systems with Applications* 36, 510–518 (2009).
13. Sheng, B., Gengxin, S.: Data mining in census data with CART. In: *Procs. of 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, pp. 260–264. Coll. of Inf. Sci. & Eng., Qingdao Univ., Qingdao, China (2010).
14. Breiman, L., Friedman, J., Olshen, R., Stone, C.: *Classification and regression trees*. Wadsworth, Monterey (1984).

15. Deichmann, J.I., Eshghi, A., Haughton, D., Woolford, S.: Measuring the International Digital Divide: an Application of Kohonen Self-organising Maps. *International Journal on Knowledge and Learning*. 3(6), 552–575 (2007).
16. Coria, S.R., Mondragón-Becerra, R., Pérez-Meza, M., Ramírez-Vásquez, S.K., Martínez-Peláez, R., Barragán-López, D., Ávila-Barrón, O. CT4RDD: Classification Trees for Research on Digital Divide. *Expert Systems with Applications* (2013). DOI: 10.1016/j.eswa.2013.04.002. In press.
17. Instituto Nacional de Estadística y Geografía (INEGI). Base de Datos por Localidad del Censo Nacional de Población y Vivienda 2010. Accesado 01/Abr/2013. <http://www.censo2010.org.mx>
18. Instituto Nacional de Estadística y Geografía (INEGI). Conformación de la Base de Datos por Localidad del Censo Nacional de Población y Vivienda 2010. Accesado 01/Abr/2013. <http://www.censo2010.org.mx>
19. Sturges, H.A. The Choice of a Class Interval. *Journal of the American Statistical Association* 21(153), 65–66 (1926).
20. Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*. 20, 37–46 (1960).

Optimización de recurso para el tratamiento de grandes volúmenes de datos

Mónica Santibáñez¹, Rosa María Valdovinos², Erendira Rendón³,
Roberto Alejo⁴ y J. Raymundo Marcial-Romero²

¹Centro Universitario UAEM Texcoco, Jardín Zumpango s/n Fracc. El Tejocote,
México

²Universidad Autónoma del Estado de México. Facultad de Ingeniería,
Cerro de Coatepec s/n Toluca, México

³Instituto Tecnológico de Toluca, Av. Tecnológico s/n Fracc. La Virgen, Metepec,
México

⁴Tecnológico de Estudios Superiores de Jocotitlán, Carretera a Toluca Atlacomulco Km. 44.8
s/n Jocotitlan, México

{monicass_isc@hotmail.com, li_rmvvr@hotmail.com,
erendon@ittoluca.edu.mx,
ralejoll@hotmail.com, jrmarcialr@uaemex.mx

Resumen El crecimiento exponencial de los datos en los últimos años ha generado la búsqueda de nuevas soluciones que permitan su tratamiento, con los recursos disponibles de forma rápida y precisa. En minería de datos esta situación es fundamental por utilizar grandes volúmenes de datos en sus procesos. En este artículo se propone una metodología centrada en la administración dinámica de la memoria RAM que permite realizar el aprendizaje *on-line* de patrones. Con la utilización de métodos de agrupamiento y clasificación con la regla de los k -vecinos más cercanos, se valida la propuesta sobre 3 conjuntos de datos, los cuales muestran la competitividad de la metodología respecto a una situación donde el conjunto de datos fuera manipulado *off-line*, con una reducción significativa en tiempo de procesamiento.

Palabras clave: aprendizaje *online*, escalabilidad, minería de datos, *clustering*.

1. Introducción

La última década ha representado, para la tecnología, una de las épocas con mayor avance, con lo que el crecimiento de los datos y el flujo de información ha sido un tema que cada vez cobra mayor interés, ya que el tratamiento de los datos se traduce en la adquisición de conocimiento. En consecuencia, el incremento exponencial de éstos, demanda que el almacenamiento y manejo de los mismos ofrezcan una respuesta en tiempo real de forma dinámica. Esta situación ha propiciado la búsqueda de nuevas soluciones que permitan el tratamiento de grandes volúmenes de datos, considerando los recursos disponibles.

Al respecto, la minería de datos es una de las áreas de la Inteligencia Artificial que se ha preocupado por proporcionar estrategias en el tratamiento de grandes volúmenes de datos para su procesamiento y adquisición de conocimiento [1]. En específico, la etapa de Reconocimiento de Patrones (RP), centrada en el reconocimiento y clasificación de objetos definidos por sus atributos, ofrece una alternativa en la adquisición de conocimiento a partir de un conjunto de datos (CD). En la aplicación del RP pueden distinguirse dos aproximaciones: aprendizaje *on-line* y aprendizaje *off-line* [2]. Por un lado, el aprendizaje *on-line* contempla el aprendizaje dinámico de patrones, utilizando para ello CD actualizados, en tanto que en el aprendizaje *off-line* se realiza un aprendizaje estático que no facilita la integración de nuevos ejemplos al conjunto de entrenamiento (CE), donde cualquier caso nuevo debe ser integrado, por un experto en el área, en un nuevo CE e iniciar nuevamente el proceso de entrenamiento.

Un aspecto importante a considerar es que muchos de los datos que se generan en tiempo real, carecen de cualquier descripción que indique de forma clara su clasificación. Para esto, las propuestas existentes se orientan a la utilización de algoritmos de agrupamiento o *clustering*, con los cuales es posible generar grupos a partir de datos sin ninguna descripción, cuyos miembros sean lo más parecidos entre sí y lo más diferentes a los de otros grupos, haciendo posible de esta forma encontrar una relación entre los datos [3].

Para que esta propuesta sea viable con grandes volúmenes de datos, es fundamental administrar de forma adecuada los recursos computacionales disponibles. El problema principal en este sentido es que en ocasiones el CD es más grande que el tamaño de memoria disponible para su procesamiento, por lo que es preciso encontrar una forma de tratarlos sin alterar su precisión. Algunas estrategias se orientan a la escalabilidad de algoritmos [4] y otras a la optimización de la memoria [5]. La primera estrategia busca mantener el adecuado funcionamiento del algoritmo de minería de datos al particionar el CD, en tanto que la segunda, busca el procesamiento de los datos atendiendo a los recursos disponibles del equipo en uso.

En el presente artículo se propone una estrategia que combina ambos enfoques, por un lado realizar la administración de la memoria RAM para trabajar con CD de gran tamaño, sobre los cuales se escalan algoritmos de agrupamiento para llevar a cabo el aprendizaje *on-line* con igual o mejor precisión que si se estuviese trabajando con el CD completo.

La estructura del artículo es la siguiente: en la sección 2 se presenta la fundamentación teórica del aprendizaje *on-line* y se describe el algoritmo de agrupamiento utilizado, en tanto que la sección 3 muestra el funcionamiento y administración de la memoria RAM en Java. La estrategia de solución se describe en la sección 4 y el análisis experimental en la sección 5. Por último se incluye una sección para las conclusiones y las líneas abiertas de estudio.

2. Aprendizaje *Online*

Actualmente, el proceso de minería de datos aplicado a grandes volúmenes de datos para obtener conocimiento requiere de técnicas capaces de soportar el procesa-

miento y análisis en tiempo real de datos; es el aprendizaje on-line o incremental, quien cubre estas necesidades, pues con su implementación se procesan flujos continuos de datos, con la capacidad de dar una respuesta en todo momento [6]. Esta aproximación resulta ventajosa principalmente cuando se requiere mantener un conocimiento del clasificador actualizado y dinámico, no obstante, la complejidad de implementación es una tarea no trivial de resolver. En términos generales el proceso de aprendizaje on-line consiste en procesar uno a uno los patrones que son ingresados [7]. De acuerdo a [8], el aprendizaje on-line supera notablemente al aprendizaje off-line (*batch*), pues a través del uso de una escala para medir el tiempo de cálculo, demuestra que el rendimiento sufre una variación significativa en varios casos.

El aprendizaje *on-line* ha sido aplicado en diversas áreas y tareas en las que la generación y el análisis de datos continuos es de gran importancia. Un caso particular es la utilización en áreas donde el procesamiento constante de cada dato aporta al sistema la información para mantenerse actualizado y poder brindar respuesta en tiempo real, como el monitoreo de redes, los sistemas de telecomunicación, los flujos generados por los clics de los clientes, los mercados de valores, entre otras tareas [9].

Dado que muchos de los datos generados en tiempo real no cuentan con ninguna descripción, es necesario contar con algoritmos que permitan identificar algún comportamiento en ellos. Los algoritmos que brindan esta posibilidad son los de agrupamiento o *clustering*. La idea general de estos métodos es separar los patrones en grupos con características similares, utilizando para ello medidas de similitud o disimilitud que aseguren la mayor semejanza entre los patrones que forman un grupo y lo más diferente a los contenidos en otros grupos [10].

En su aplicación se pueden distinguir dos enfoques: el *hard clustering*, en el cual se generan grupos sin solapamiento y que a su vez se divide en algoritmos de partición, que generan un determinado número de grupos dependiendo de la función de densidad y los algoritmos jerárquicos, que generan agrupamientos internos a uno existente, y el *soft clustering*, que trabaja con el solapamiento de grupos. Estos algoritmos están basados en los métodos difusos, las redes neuronales artificiales y los algoritmos genéticos.

En esta propuesta se emplea el algoritmo de *clustering* Batchelor y Wilkins, correspondiente al *hard clustering*, también conocido como algoritmo de máxima distancia. En su funcionamiento requiere del parámetro θ , el cual, de acuerdo a la literatura, debe estar en el rango $0 \leq \theta \leq 1$, e indica la distancia media entre los grupos existentes. A partir del parámetro θ se calcula un umbral de distancia que permite decidir cuándo se crean nuevos grupos [11].

El algoritmo consiste en asignar arbitrariamente uno de los patrones como centro del primer grupo y utilizando la distancia, en este caso Euclídea, seleccionar el patrón más alejado y asignarlo como centro del segundo grupo. A partir de los dos grupos creados se crea una tabla que contiene los pares (patrón, centro más cercano), entre ellos se elige el patrón más alejado de los grupos existentes y se calcula el umbral de distancia, este umbral es comparado con la distancia del patrón a su centro más cercano, si el umbral es superado se crea un nuevo grupo, si no concluye este paso y se continua con la agrupación libre, donde los patrones que no fueron seleccionados

como centro de grupo de forma iterativa son asignados al centro más cercano, recalculando de esta forma el centroide, hasta que se concluye con todos los patrones [11].

3. Manejo de memoria RAM con Java

El manejo de memoria permite adaptar el recurso de RAM a las necesidades que se tengan, por ejemplo, es posible tener acceso a los datos almacenados en la RAM mediante acceso directo a través de una dirección de memoria y realizar algunas acciones como la lectura y escritura [12]. En el caso de las aplicaciones, la memoria RAM a utilizar es designada de antemano, en el caso de Java (el lenguaje de programación que se utiliza para desarrollar la propuesta), interactúa con la RAM a través de la JVM (JVM por sus siglas en inglés, Java Virtual Machine), siendo esta la responsable de que Java sea multiplataforma. En la división de memoria de la JVM se maneja el espacio que será asignado tanto a los datos permanentes, como a datos que son dinámicos, es decir, aquellos necesarios sólo durante un periodo de la ejecución (Fig. 1) [13].

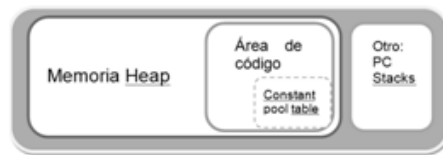


Fig. 1. Estructura de la memoria JVM.

En la estructura de la memoria de la JVM, la memoria *Heap* se crea al iniciar la máquina virtual de Java, esta se carga inicialmente con espacio de 64 MB, los cuales pueden aumentar o contraerse, de acuerdo a las necesidades de la aplicación. Dentro de esta área de memoria se implementa el administrador automático de almacenamiento (*automatic storage management*) [14] que mantiene un historial de los objetos y al momento de no ser referenciados libera el espacio con el uso del recolector de basura o *Garbage Collector*.

El área de métodos se crea con el inicio de la máquina virtual y puede también ser de tamaño fijo o dinámico, de acuerdo a los requerimientos en tiempo de ejecución, aunque también implementa un límite máximo, en este caso la memoria tampoco necesita ser contigua. Java no provee mayor información sobre la memoria, ya que estos datos se dejan a discreción del implementador, pero dentro de la información que si puede ser consultada, se encuentran las instrucciones para manipular el tamaño de las dos áreas de memoria [15]. Al respecto, una de las propuestas más recientes es la de Choi quien propone un algoritmo escalable de memoria externa, que permite tratar con el problema *MaxRs* de las bases de datos espaciales y los algoritmos *in-memory*, así mismo propone un algoritmo para el problema *MaxCRS* [5].

4. Herramientas y métodos

En esta sección se describen las herramientas y métodos utilizados en la propuesta, donde se busca realizar una recuperación eficiente de patrones en forma on-line y la aplicación de escalabilidad al algoritmo de agrupamiento.

4.1. Administración de memoria

Para identificar la capacidad de memoria RAM del equipo en uso, y a partir de ésta determinar el tamaño que los subconjuntos han de tener para ser procesados de forma consecutiva, se utiliza la librería Libsigar [16], que implementa las funciones necesarias para conocer el tamaño total de memoria en MB, así como el número de procesadores con los que cuenta el equipo, cantidad de espacio libre de memoria, que es manejada en KB, entre otros datos del sistema. Al trabajar con Java, puede modificarse el tamaño por default asignado a la memoria, para aumentar la capacidad y evitar desbordamientos.

Una vez identificado el recurso disponible, se destina un 10% para la ejecución de la aplicación. De acuerdo a este espacio disponible se asigna el 80% a la memoria volátil de la JVM y el 20% a la memoria permanente. Esto se debe a que en la memoria volátil se almacenan las variables dinámicas, que serán utilizadas por un “periodo”, mientras que la memoria permanente almacena aquellos datos que estarán presentes durante toda la ejecución del programa, entre ellas se encuentran las librerías, clases, interfaces, entre otras.

Para fines de experimentación con los conjuntos de datos obtenidos, el tamaño fijado en la simulación para la memoria, se realiza a partir del tamaño del conjunto de datos en Bytes. Si el tamaño del conjunto en Bytes no es un número entero entonces este se redondea al siguiente entero, ya que para calcular el número de Bytes cargados en memoria se manejan sólo cantidades enteras. Enseguida la cantidad obtenida como tamaño del conjunto de datos es dividida entre el número de conjuntos con los que se desea realizar la prueba, a esta cantidad se suma el número de patrones existentes en el conjunto, ya que a cada línea hay que agregar el espacio que ocupa el carácter de salto de línea. De esta forma es posible especificar en los experimentos el tamaño total de RAM.

Para el caso particular de los experimentos aquí realizados se utilizó una computadora Dell Inspiron 1545 con 3 GB en RAM, procesador Intel® Core™ 2 Duo T6600 a 2.20 GHz, Disco Duro de 250 GB, con sistema operativo Windows 7 de 64 bits.

4.2. Conjuntos de datos

Los conjuntos de datos fueron tomados del repositorio de la SIPU (por sus siglas en inglés *Speech and Image Processing Unit*) de la Universidad del Este de Finlandia [17]. La Tabla 1 muestra las características de los CD utilizados. La primera columna corresponde a un conjunto artificial de formas arbitrarias, D31, en la segunda y terce-

ra columna se muestra una descripción de los CD reales que corresponden a ubicaciones de sitios de interés generados por usuarios.

Table 1. Conjunto de ubicaciones de usuarios

Característica	D31	Joensuu	MOPSI
No. de vectores	3100	6014	8589
No. de <i>clusters</i>	31	-	-
Dimensiones	2	2	2
Tamaño de archivo	49.5KB	110KB	155KB

Pese al tamaño de los CD, no fue posible conseguir uno que de forma real superara los recursos disponibles de RAM. Por esta situación, se introdujo a la aplicación un valor que simula que el tamaño de la memoria es menor al real, para poder trabajar con la división del conjunto de datos de acuerdo a la capacidad de RAM.

Cada uno de los CD se divide de acuerdo al tamaño asignado a la memoria volátil de la JVM. Por lo tanto se obtienen n subconjuntos de un tamaño no mayor al especificado. Cada uno de estos subconjuntos es procesado de forma iterativa hasta concluir con todo el CD. Los subconjuntos obtenidos para todos los conjuntos de datos fueron tres y los tamaños correspondientes son: D31=19.5312KB, Joensuu=48.4131KB y MOPSI=52.2461KB. Con los subconjuntos obtenidos se aplicó el algoritmo heurístico incremental para encontrar los grupos de datos que comparten características similares. Como ya se explicó en la sección 2, el algoritmo requiere del parámetro libre θ . Para fines de esta experimentación, los valores de los parámetros libres son: $\theta = 0.1, 0.2, 0.3, 0.5, 0.7, 0.9$.

5. Resultados y discusión

Para fines de validación, a cada subconjunto obtenido se aplicó el método de validación cruzada [18] con 2 repeticiones, utilizando un 80% de los patrones para entrenamiento y el 20% restante para el conjunto de prueba, en tanto que para analizar los resultados se utilizó la precisión general, expresada por la siguiente ecuación (1).

$$PG = \frac{\sum e^x}{p} \quad (1)$$

donde e = patrones correctamente clasificados y p = total de patrones presentados para su clasificación.

5.1. Clasificación

La Figura 2 muestra los resultados obtenidos al aplicar el algoritmo Batchelor y Wilkins, seleccionado por ser un algoritmo que no requiere conocer de antemano el número de grupos que deben ser encontrados, además de contar sólo con un parámetro libre que permite calcular de forma dinámica el umbral para saber si debe crear un nuevo grupo o no. Como valor de referencia, se incluyeron los resultados obtenidos

con el CD original, es decir ejecutando el algoritmo de agrupamiento sobre el CD cargado en su totalidad en la RAM. El algoritmo de clasificación base utilizado es la regla del vecino más cercano en su modalidad k-NN [19], siendo $k=3$ para todos los conjuntos de forma completa y on-line, ya que este es el valor que obtuvo la precisión más alta procesando el conjunto real con mas instancias de forma completa.

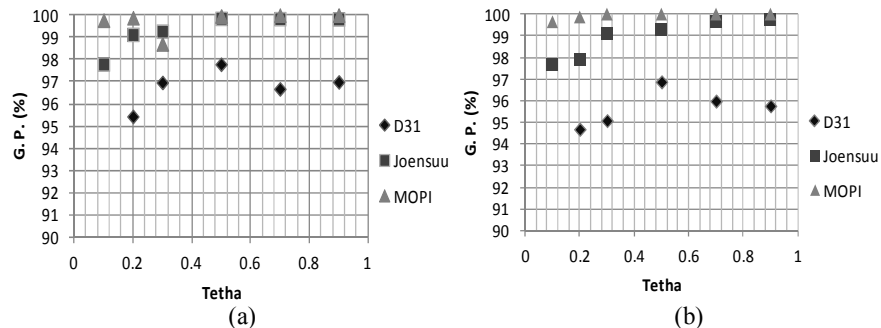


Fig. 2. Precisión general obtenida al procesar el conjunto con el método de *clustering* y la regla k-NN, para cada uno de los conjuntos de datos. (a) precisión para el conjunto completo, (b) la precisión de forma *on-line*.

En ambas gráficas de la Figura 2, se puede observar la precisión general obtenida para cada valor de θ , donde el eje X representa el valor del parámetro θ , y el eje Y el valor de la precisión general, finalmente, la serie indica el conjunto de datos procesado. En la Figura 2a puede observarse la PG obtenida para los conjuntos de datos completos, y en la Figura 2b el resultado del procesamiento por subconjuntos.

En estos resultados es posible observar varios aspectos, primeramente las precisiones obtenidas tanto con la metodología de procesamiento on-line, como las obtenidas utilizando el CD completo, sin importar el valor que tome θ son muy parecidas. En el caso del conjunto D31 procesado de forma completa se tiene una precisión general de 95.09%, mientras que al realizar el proceso on-line es de 93.43%. Para el conjunto Joensuu la precisión que se obtiene con el conjunto completo es en promedio del 99.28, y on-line es del 98.90%. Por último, el procesamiento on-line supera con 99.91% la precisión obtenida con el CD completo, que es de 99.72%.

Por otro lado, respecto al rendimiento del algoritmo de *clustering*, es posible observar que los valores de θ que mejor rendimiento ofrecen al procesar el CD completo son de 0.5 a 0.9, en tanto que para el aprendizaje on-line en su mayoría es de 0.7. Un caso particular son los resultados obtenidos con el CD MOPSI, pues con cualquier valor de θ ofrece una precisión entre 99% y 100%.

Finalmente, cuando se asigna el valor de $\theta = 0.1$, la precisión es menor con respecto a los demás valores. Esta situación puede observarse con claridad en el conjunto D31, cuya precisión general para dicho valor oscila entre 71 y 86%.

5.2. Tiempo de procesamiento

El tiempo requerido en el procesamiento de los CD es un factor importante a analizar, ya que la velocidad en procesamiento es uno de los objetivos que, adicional a la precisión, son perseguidos en minería de datos. En este sentido, el análisis se orienta a validar la propuesta de realizar la escalabilidad del algoritmo de *clustering* al realizar el aprendizaje on-line, respecto a la utilización del CD en su totalidad cargado en memoria.

La comparativa de los resultados de tiempo del tratamiento de los conjuntos se observan en la Figura 3. Esta figura muestra dos gráficas en las que en el eje de las X se incluyen los valores de θ , en tanto que en el eje de las Y se tiene el tiempo requerido para el procesamiento de cada uno de los CD utilizados. Este tiempo es medido en segundos.

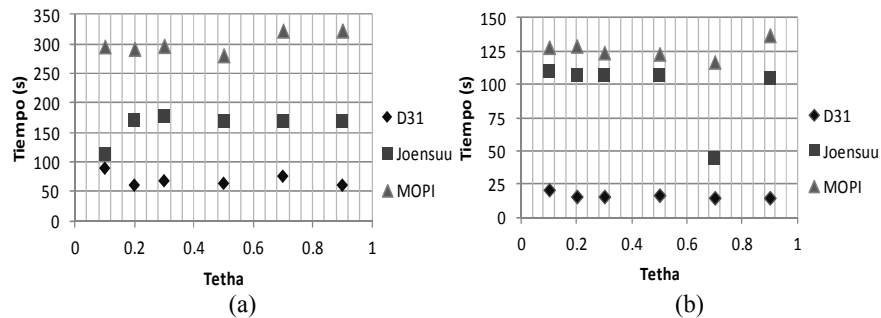


Fig. 3. Tiempo de procesamiento obtenido con cada uno de los CD. (a) Muestra la gráfica para el CD completo, (b) El tiempo por aprendizaje *on-line*.

En estos resultados es interesante observar que al realizar el procesamiento de los CD con aprendizaje on-line, el tiempo es significativamente menor respecto a cuándo se procesa el CD completo en memoria. Esta situación es sumamente ventajosa ya que al realizar la administración de la memoria se tiene un ahorro hasta del 50% con un rendimiento similar a si el CD hubiese estado residente en su totalidad en la RAM (Fig. 2).

Respecto a cada uno de los CD, es posible observar que el CD D31 es 22% más rápido de forma on-line que cuando se procesa el CD de forma completa. El conjunto Joensuu se procesa en promedio un 38% más rápido que de forma completa y el conjunto MOPSI procesado de forma on-line tuvo una disminución promedio del 58% de tiempo en contraposición a su tratamiento de forma completa.

El resultado de tiempo de procesamiento está estrechamente ligado a los parámetros libres del algoritmo de *clustering*. En este sentido, es posible notar que no siempre el valor de θ que arroja la mejor precisión procesa también el conjunto en menor tiempo. Cuando se procesa el CD completo, excepto con el conjunto Joensuu la mejor precisión y el menor tiempo son obtenidos con diferente valor de θ (0.5 y 0.2 respectivamente). Esta misma situación se presenta al realizar el procesamiento on-line, el CD que proporciona un mejor rendimiento con el mismo valor de θ en precisión y

tiempo, es MOPSI ($\theta = 0.3$). No obstante, estas diferencias son poco significativas ya que la precisión obtenida en su equivalente con el CD completo es muy similar en todos los casos. Por último, para todos los conjuntos de datos la diferencia entre los tiempos con respecto al mejor, oscila entre 5 y 15 segundos.

6. Conclusiones

En este artículo se propone la administración de la memoria para permitir la realización del aprendizaje on-line de patrones, dos aspectos fueron analizados, el rendimiento del clasificador y el tiempo de preprocesamiento. Los resultados obtenidos, muestran que, tanto en la simulación del aprendizaje on-line, como la utilización del CD residente en su totalidad en memoria, la precisión es muy similar. No obstante, la reducción en el tiempo requerido por el aprendizaje on-line, es significativamente inferior, siendo en la mayoría de los casos hasta en un 50%.

Esta situación podría estar ocasionada por la cantidad de patrones a procesar en cada momento, en el aprendizaje *on-line* es menor. Esto tiene que ver no sólo con la memoria, sino también con la cantidad de procesos que realiza el procesador. Por tanto, la administración de memoria y la adaptación de los grandes CD hacen que puedan ser procesados en menor de tiempo, sin perder precisión y aprovechando al máximo el recurso computacional. Respecto a los valores de θ se pudo observar que los mejores resultados se obtienen cuando su valor oscila entre 0.3 y 0.9.

Las líneas abiertas de estudio se orientan a ampliar la experimentación con otros CD, buscando la adquisición de CD que superen de manera real el recurso computacional disponible. Además de analizar la conveniencia de utilizar los agrupamientos realizados de forma *on-line* para construir *ensembles* de clasificadores.

Agradecimientos. Este trabajo fue realizado gracias al apoyo recibido del proyecto de la UAEM Generación y conteo de cubiertas de aristas en graficas acíclicas.

Referencias

1. Hernández O., J., Ramírez Q., M. J., Ferri R., C.: Introducción a la Minería de Datos. Person Educación, Madrid (2004)
2. Alpaydin, E.: *Introduction to Machine Learning* (2ª ed.). The MIT Press, Cambridge, MA (2010)
3. Aggarwal, C.: Framework for Clustering Massive-Domain Data Streams. En: IEEE 25th International Conference on Data Engineering, pp. 102-113. IEEE Press, New York (2009)
4. Khalilian, M., Mustapha, N.: Data Stream Clustering: Challenges and Issues. En: Proceedings of the International MultiConference of Engineers and Computer Scientists 1 (2010)
5. Choi, D.W., Chung, C.W., Tao, Y.: A Scalable Algorithm for Maximizing Range Sum in Spatial databases. Proceedings of the VLDB Endowment, 5th edn. (2012)
6. Ferrer T., F. J., Aguilar R., J. S.: Aprendizaje Incremental de Reglas en Data Streams. Actas del III Taller Nacional de Minería de Datos y Aprendizaje, TAMIDA, pp. 261-270. Thomson, Sevilla (2005)

7. Mathieu, C., Sankur, O.: Online correlation clustering. Symposium on Theoretical Aspects of Computer Science, 573-584 (2010)
8. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online Dictionary Learning for Sparse Coding. Proceedings of the 26th International Conference on Machine Learning. (2009)
9. Beringer, J., Hullermeyer E.: Online clustering of data streams. Technical Report 31, Department of Mathematics and Computer Science, Philipps-University Marburg, Germany (2003)
10. Marques de Sá P., J.: Pattern Recognition, Concepts, Methods and Applications. Springer, Alemania (2001)
11. Murty, M. N., & Devi, V. S.: Pattern recognition: An algorithmic approach. Springer (2012)
12. Cortijo, B. F. J.: Técnicas no supervisadas: Métodos de agrupamiento. Consultado en septiembre de 2012. Apuntes disponibles en la web de la asignatura [<http://www-etsi2.ugr.es:8080/depar/ccia/RF0708/material.htm>]
13. Lindhom, T., Yellin, F., Bracha, G., Buckley, A.: The Java Virtual Machine Specification. Java SE 7 Edition. Oracle America, Estados Unidos (2011)
14. Menchaca, M. R., García C. F. (2000). Arquitectura de la Máquina Virtual Java. Revista Digital Universitaria 1(2). Consultado en septiembre 2012, <http://www.revista.unam.mx/vol.1/num2/art4/>
15. Sun, Microsystems (2006). Memory Management in the Java HotSpot Virtual Machine. Sun Microsystems Inc.
16. Hyperic HQ, <http://support.hyperic.com/display/SIGAR/Home>
17. Speech and Image Processing Unit, Clustering datasets, <http://cs.joensuu.fi/sipu/datasets/>
18. Theodoridis, S., Koutroumbas K.: Pattern Recognition (4th ed.). Elsevier Academic Press, Estados Unidos (2006)
19. Dasarathy, B.V.: Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamos, CA, (1991)

Sistema de enseñanza para la técnica de clasificación de árboles de decisión

A. Franco-Arcega, F.A. Castro-Espinoza y P. Cortes-García

Área Académica de Computación
Universidad Autónoma del Estado de Hidalgo
Carretera Pachuca-Tulancingo Km. 4.5, C.U.
Pachuca, Hidalgo, México, C.P. 46600

Resumen Los Árboles de Decisión son una de las técnicas de clasificación supervisada más utilizada para resolver problemas. Por este motivo, es necesario que usuarios, principalmente de las áreas relacionadas a computación, comprendan el procedimiento que se lleva a cabo para construir un Árbol, con la finalidad de poder aplicar este tipo de técnicas en la solución de problemas cotidianos. Actualmente, se ha detectado que el transmitir este conocimiento utilizando técnicas de enseñanza tradicionales no es una tarea fácil, ya que en la definición de un problema real puede existir una gran cantidad de atributos que lo describa, lo que conlleva a tener más operaciones matemáticas para generar un Árbol. En particular, en este trabajo se propone automatizar la herramienta de estudio sobre la técnica de clasificación supervisada denominada Árboles de Decisión desarrollando un sistema de enseñanza computacional. Este sistema tendrá los objetivos de mostrar cómo se construye un Árbol de Decisión paso a paso, analizando cada una de las operaciones matemáticas realizadas para tal fin.

Palabras clave: sistema de enseñanza, árboles de decisión, minería de datos.

1. Introducción

En la actualidad, la tecnología computacional ayuda a resolver problemas y satisfacer necesidades cotidianas. El ámbito académico es uno de los principales entornos en donde la tecnología introduce sus beneficios. Expertos de la computación se han interesado en brindar a los alumnos material académico que los lleve de la mano para la comprensión de ciertos temas complejos. Algunos trabajos [1,2,3] han propuesto diferentes herramientas para la comprensión de temas relacionados al procesamiento de datos con técnicas de computación inteligente.

En materias como Minería de Datos, Reconocimiento de Patrones o Inteligencia Artificial se trabaja con este tipo de técnicas inteligentes, con las cuales se resuelven problemas de predicción o descripción [4]. Los Árboles de Decisión son una de las herramientas más utilizadas para darle solución a problemas de tipo predictivo. Un Árbol de Decisión es una técnica que ayuda a tomar decisiones

adecuadas entre muchas posibilidades, aproxima funciones de valores discretos de destino y puede ser representado como un conjunto de reglas para mejorar la legibilidad humana. Estos métodos de aprendizaje se encuentran entre los más populares de los algoritmos de inferencia inductiva y se han aplicado con éxito a una amplia gama de tareas [5]. Un Árbol de Decisión es una estructura que contiene nodos y ramas. Los nodos de un árbol pueden ser de dos tipos: internos o hojas. Los nodos internos se caracterizan por tener uno o más atributos de prueba, los cuales ayudan a decidir qué camino es más satisfactorio para recorrer el árbol. Por su parte, los nodos hoja contienen etiquetas de clase, las cuales definen la decisión a tomar.

Uno de los problemas que se detecta en el ámbito académico es que no es fácil transmitir el conocimiento de este tipo utilizando técnicas de enseñanza tradicionales, como papel y lápiz, y por consecuencia es difícil de comprender por los alumnos. Mientras más atributos están involucrados en la definición del problema, más operaciones matemáticas son utilizadas para la generación de un Árbol de Decisión y por tanto para una efectiva toma de decisiones. Por esta razón, se han desarrollado diversos sistemas que de manera automática ejecutan todas las operaciones matemáticas para llevar a cabo el procedimiento de alguna técnica en específico. En particular, en este trabajo se propone desarrollar un sistema de enseñanza de la Técnica de Árboles de Decisión, con el fin de que el alumno pueda apreciar cómo se construye un árbol paso a paso, analizando cada una de las operaciones matemáticas realizadas.

El documento está organizado de la siguiente forma: en la sección 2 se describen los principales sistemas que permiten la creación de Árboles de Decisión. En el apartado 3 se describe el sistema propuesto en el presente artículo detallando sus funcionalidades y las etapas de creación de un Árbol de Decisión. En la sección 4 se presentan los resultados preliminares del sistema propuesto. Finalmente, en el apartado 5 se presentan las conclusiones y trabajos futuros del presente trabajo de investigación.

2. Trabajos relacionados

Actualmente, se han desarrollado algunos sistemas que permiten la construcción de Árboles de Decisión. Sin embargo, ninguno de estos sistemas muestra paso a paso el proceso que se lleva a cabo para esta generación, ya que en los sistemas simplemente se visualiza como queda el árbol final. Al usuario no se le presenta el procedimiento de construcción, por lo cual no podrá comprender que operaciones matemáticas se realizaron para llevar a cabo el proceso.

Precision Tree es un sistema que permite desarrollar Árboles de Decisión que proveen una estructura formal en la que las decisiones y los acontecimientos casuales están relacionados en secuencia de izquierda a derecha. Las decisiones, los eventos al azar, y los resultados finales se representan mediante nodos conectados por ramas [6]. El resultado es una estructura de Árbol con la raíz en la izquierda y los caminos diversos a la derecha. Aunque visualmente se entiende

una decisión, ya que identifica las mejores opciones y comunica los resultados, el proceso de construcción no se muestra paso a paso.

El sistema GATree hace uso de algoritmos genéticos para evolucionar o construir árboles binarios de Decisión que ajuste al concepto de destino. Por ello GATree adopta una representación natural del espacio de búsqueda usando Árboles de Decisión reales y no de cadenas binarias [7]. GATree es un constructor de Árboles de Decisión que se basa en algoritmos genéticos, el cual asume que si se tienen amplios recursos, entonces se podrá esperar un árbol cada vez de mejor ajuste. GATree puede proporcionar un conjunto de Árboles de Decisión que son totalmente diferentes, pero que son opciones cercanas al espacio de solución. Al igual que el sistema anterior, GATree no permite al usuario visualizar el proceso de construcción, simplemente muestra el árbol terminado.

SMILES es un sistema de aprendizaje automático que integra muchas características diferentes de las técnicas de otras máquinas y paradigmas de aprendizaje. En particular, SMILES tiene un manejo de los recursos sofisticado y muy eficaz. De esta manera, SMILES combina y mejora el reciente interés en combinación de hipótesis sensibles a los costes de aprendizaje [8]. Sus aplicaciones son básicamente para la Minería de Datos y cualquier otra tarea de aprendizaje donde los Árboles de Decisión podrían ser útiles. Este sistema aunque es muy robusto en cuanto al manejo de los recursos, no es apropiado para la enseñanza de la técnica de árboles, ya que el usuario no comprendería todas las características de él, además de que no observaría de manera detallada la generación de un árbol.

See5 es una sofisticada herramienta de Minería de Datos para descubrir patrones que definen categorías, los cuales pueden ser utilizados para hacer predicciones [9]. See5 ha sido diseñado para analizar bases de datos sustanciales que contienen miles de millones de registros y decenas a cientos de atributos de tipo numérico, hora, fecha o campos nominales. Para maximizar la interpretabilidad, See5 expresa sus resultados como Árboles de Decisión o como conjuntos de reglas if-then. Aunque See5 es fácil de usar y no supone ningún conocimiento especial de Estadística o Machine Learning, no le permite al usuario entender el proceso interno que se lleva a cabo para llegar al resultado final.

Angoss es un software de análisis predictivo que ayuda a las empresas a descubrir información valiosa en sus datos, como el descubrimiento de oportunidades para aumentar las ventas y la rentabilidad, y reducir el riesgo [10]. Angoss permite procesar atributos discretos (categóricos) y continuos. Además, el usuario puede definir algunas preferencias en el crecimiento del Árbol (por ejemplo, árboles binarios o n-arios), tipos de valor de agrupación y los intervalos abiertos. Este sistema muestra un informe acerca de las reglas de cada nodo y las estadísticas para cada decisión (nodos hoja). Debido a que Angoss está enfocado a la solución de análisis en empresas, no es viable para ser utilizado en la enseñanza de esta técnica de predicción.

Weka es una herramienta desarrollada en lenguaje Java por la Universidad de Waikato, Hamilton, New Zeland [11]. Weka permite la extracción de conocimiento desde bases de datos y debido a que fue desarrollado bajo licencia GNU-

GPL es una de las suites más utilizadas actualmente en las áreas de aprendizaje automático y minería de datos. Esta herramienta permite al usuario aplicar diversos algoritmos de clasificación, agrupamiento, selección de variables, entre otros. Sin embargo, Weka no muestra al usuario como se lleva a cabo el procesamiento de las bases de datos, únicamente permite cargar en memoria la base de datos y elegir el algoritmo a aplicar, para después sólo mostrar el resultado final del procesamiento.

3. Sistema de enseñanza propuesto

Los Árboles de Decisión son una herramienta valiosa que permite solucionar diversos problemas de clasificación supervisada o predicción. Para que un usuario pueda saber como aplicar esta herramienta en alguna situación real, es necesario que comprenda su funcionamiento. Un Árbol de Decisión ayuda a la toma de decisiones efectivas, ya que sus nodos hoja contienen etiquetas de clase que son asignadas a un nuevo caso. Es importante que el usuario entienda como se construye un árbol para que visualice los patrones que ayudan en la toma de decisiones, así como para observar que atributos son importantes en su problema.

El Sistema propuesto, denominado SETECAD, muestra paso a paso la construcción de un Árbol de Decisión tomando en cuenta un orden jerarquizado, es decir, por niveles de nodos, los cuales se identificarán fácilmente. Además, las operaciones matemáticas que son utilizadas para expandir los nodos internos, también se muestran en una ventana del sistema.

El sistema que permite enseñar la técnica de Clasificación Supervisada de Árboles de Decisión fue desarrollado en lenguaje Java, utilizando elementos gráficos y de dibujo de este lenguaje. Se tiene una interfaz principal y diferentes ventanas que muestran variados elementos que sirven para la comprensión del proceso de construcción de un Árbol de Decisión. Estas ventanas son: Lectura de datos, creación de un Árbol de Decisión, desarrollo computacional, formulario, ayuda e impresión. En las siguientes subsecciones se describen cada una de las funcionalidades del sistema SETECAD. La figura 1 muestra la interfaz principal del sistema SETECAD.

3.1. Lectura de datos

Esta ventana permite abrir el archivo que contiene los datos que van a ser utilizados para la construcción del Árbol de Decisión, el cual debe contener como primera línea el nombre de los atributos de dichos datos. Si el conjunto de datos está descrito por n atributos, las n -líneas siguientes serán destinadas para la declaración del tipo de atributos y clase, es decir, los atributos categóricos se declararán con los posibles valores de dicho atributo, los atributos de tipo continuo se declaran con la palabra reservada `#NUMERO` y la clase con la palabra `CLASE` antecedido de las etiquetas de clases en dichos datos. La Figura 2 muestra un ejemplo de un documento de entrada que contiene un conjunto de datos. Una vez cargada la información se almacena en memoria y se detalla en pantalla

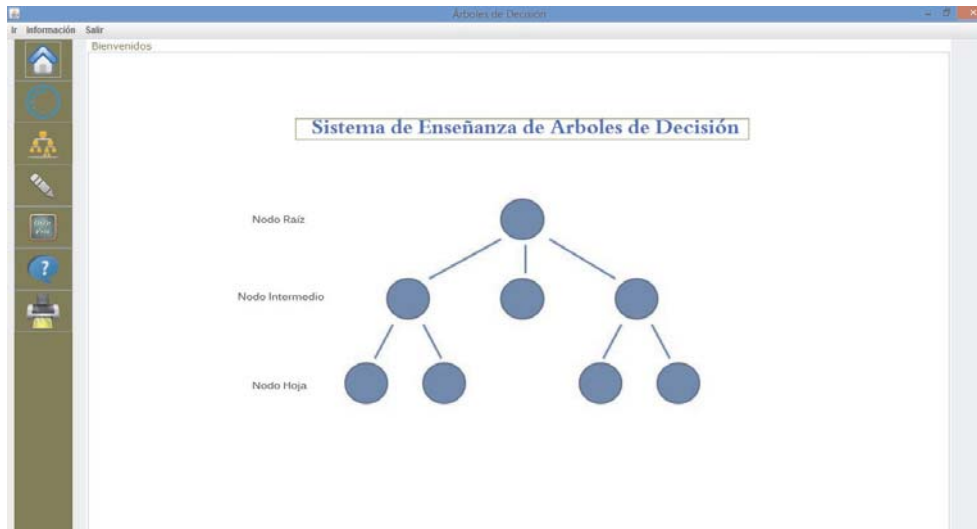


Figura 1. Interfaz principal del sistema SETECAD

el contenido de esta información, en específico, se muestra el número total de objetos contenidos en el archivo, el número de atributos y si dichos atributos son categóricos o continuos, el número de clases y finalmente, se muestra una porción de los datos antes almacenados. Esto se muestra en la figura 3

3.2. Creación de un árbol de decisión

En esta ventana se construye gráficamente el Árbol, paralelamente a los cálculos que se realizan en la ventana de Desarrollo Computacional, es decir, conforme se van realizando las iteraciones se va actualizando el gráfico que muestra el árbol construido. Esta construcción inicia en el nodo raíz, eligiendo al atributo que contenga mayor ganancia de información (ver sección 3.4), y posteriormente se dividen los objetos de entrenamiento de acuerdo a dicho atributo seleccionado. En la Figura 4 se observa cómo se grafica el nodo raíz una vez expandido, enumerando los nodos generados en el árbol de decisión. La Figura 5 muestra el árbol actualizado una vez que se ha expandido el nivel dos de éste.

3.3. Desarrollo computacional

Esta ventana imprime el procedimiento metodológico y matemático para la construcción de un Árbol de Decisión. La parte metodológica muestra los pasos que se llevan a cabo para la expansión de los nodos, especificando al usuario las instrucciones realizadas en el proceso de expansión. Además el usuario puede observar las operaciones matemáticas calculadas para tal fin. La Figura 6 muestra una parte de lo que se puede observar en esta ventana de desarrollo computacional.

```

play - Notepad
File Edit Format View Help
OUTLOOK TEM HUMIDITY WINDY CLASS
sunny overcast rain
#NUMERO
#NUMERO
T F
CLASE play dont`play
*****
sunny 75 70 T play
sunny 80 90 T don`tplay
sunny 85 85 F don`tplay
sunny 72 95 F don`tplay
sunny 69 70 F play
overcast 72 90 T play
overcast 83 78 F play
overcast 64 65 T play
overcast 81 75 F play
rain 71 80 T don`tplay
rain 65 70 T don`tplay
rain 75 80 F play
rain 68 80 F play
    
```

Figura 2. Formato del archivo

3.4. Formulario

En cada nodo interno de un Árbol de Decisión se debe seleccionar un atributo para dividir este nodo. La teoría de la información, la cual está basada en la entropía, se toma como referencia para elegir al mejor atributo en este sistema propuesto. Entre más pequeño sea el valor de la entropía, menor será la incertidumbre y más útil será el atributo para la clasificación. Para encontrar la medida en como un conjunto S es dividido de acuerdo a sus clases se suman las frecuencias de la proporción de cada clase i , siguiendo la ecuación 1.

$$\text{inf}(S) = - \sum_{j=1}^k \frac{\text{freq}(C_j, S)}{|S|} \times \log_2 \left(\frac{\text{freq}(C_j, S)}{|S|} \right) \quad (1)$$

Si se considera este proceso para encontrar la medida en cómo se particiona el conjunto S , de acuerdo a un número n de posibles valores de un atributo en específico, se calcula la suma ponderada de cada subconjunto de objetos. La ecuación 2 define esta operación.

$$\text{inf}_x(S) = \sum_{i=1}^n \frac{|S_i|}{|S|} \times \text{inf}(S_i) \quad (2)$$

Utilizando las medidas anteriores se puede obtener la ganancia de información de un atributo X , basada en la disminución de la entropía del conjunto S . La ecuación 3 muestra la fórmula de la ganancia de información.

$$\text{gain}(X) = \text{inf}(S) - \text{inf}_x(S) \quad (3)$$

Cuando un atributo contiene una gran cantidad de posibles valores, especialmente los atributos numéricos, éstos deben procesarse de una manera diferente. La ganancia de información de un atributo de este tipo debe ser normalizada de

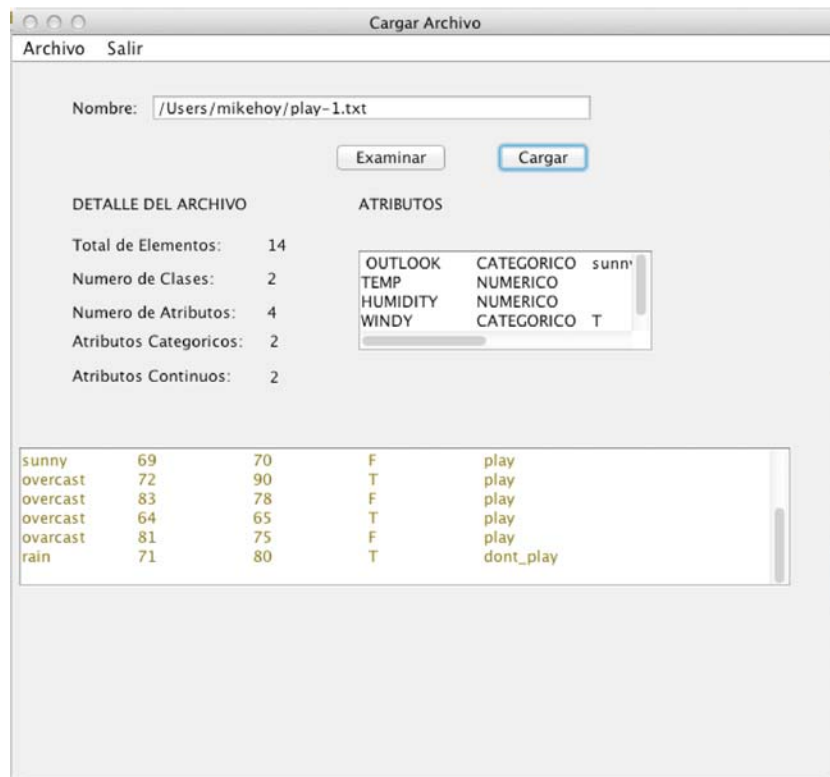


Figura 3. Ventana de lectura de datos del sistema

acuerdo al número de posibles valores que tenga el atributo. Esta normalización se hace con la ecuación 4. Finalmente, la proporción de ganancia de información (Gain Ratio) se efectúa aplicando la ecuación 5.

$$split(X) = - \sum_{i=1}^n \frac{|S_i|}{|S|} \times \log_2 \left(\frac{|S_i|}{|S|} \right) \quad (4)$$

$$ratio(X) = \frac{gain(x)}{split(x)} \quad (5)$$

El atributo con la mayor ganancia de información es el elegido para la división del nodo. Las fórmulas expuestas en esta sección se muestran en la ventana de Formulario del sistema de enseñanza propuesto.

3.5. Ayuda

Al dar clic en el botón de ayuda se abre un archivo que al usuario le permitirá saber como es el funcionamiento de dicho sistema. La Figura 7 muestra dicho documento.

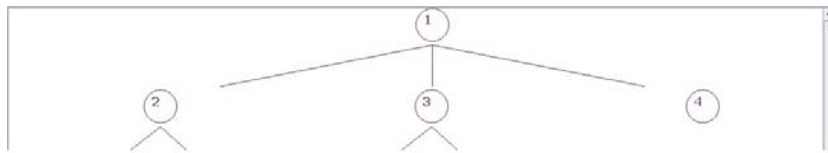


Figura 4. Construcción con un nivel

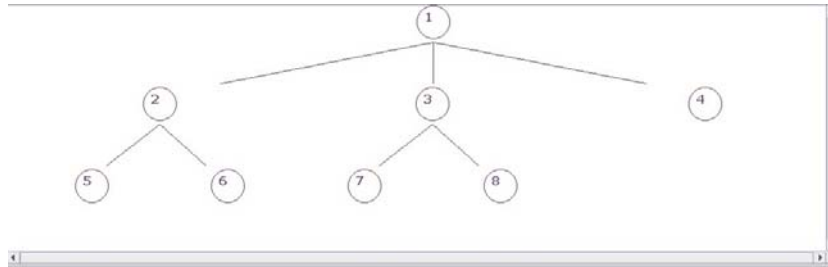


Figura 5. Construcción con dos niveles

Valor 3 :	2
Calculo con Eq. 1 :	0.0
Calculo con Eq. 2 :	6.730116670092565
Ganancia del atributo (Eq. 3) :	0.48072261929232607
Número de Atributo :	2
Tipo de Atributo :	Numérico
Paso 1.1: Ordenar los valores numéricos del atributo	
Paso 1.2: Para cada par de valores del atributo X, Y obtener el promedio entre ellos (Xp) y calcular la Ganancia	
particionando los objetos en el nodo con la condición if/else (valorAtributo <= Xp)	
Promedio (Xp) 1 :	64.5
Valores Atributo <= Xp	
Calculo con Eq. 1 :	0.0
Calculo con Eq. 2 :	0.0
Valores Atributo > Xp	
Calculo con Eq. 1 :	0.666278442414676

Figura 6. Desarrollo computacional

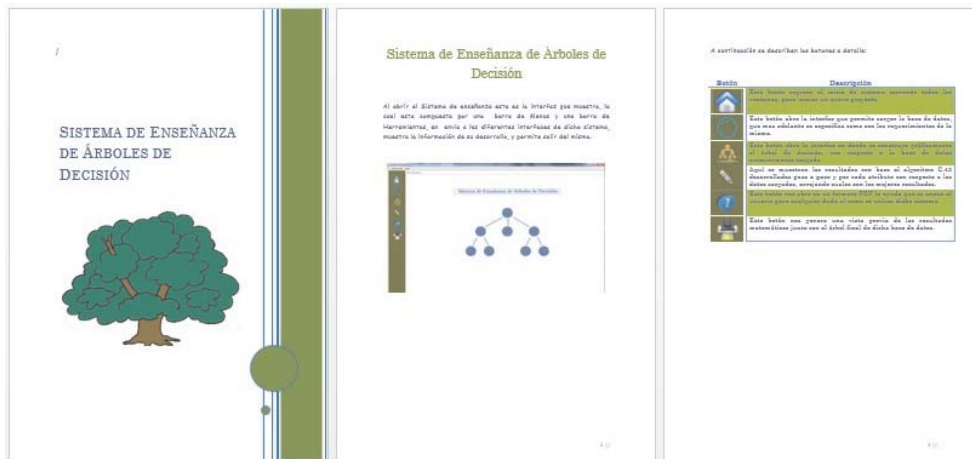


Figura 7. Manual de usuario

4. Resultados preliminares

El sistema SETECAD se les ha proporcionado, como apoyo educativo, a 5 estudiantes de la materia de Minería de Datos, de la Maestría en Ciencias Computacionales del Centro de Investigación en Tecnologías de Información y Sistemas (CITIS) de la Universidad Autónoma del Estado de Hidalgo (UAEH). Los avances académicos mostrados por los 5 estudiantes han sido importantes, en comparación con los obtenidos por otros estudiantes en anteriores versiones de la misma materia. Minería de Datos se imparte semestralmente a un promedio de entre 5 y 10 alumnos, sin embargo, estos resultados no podrán ser validados de manera experimental hasta no evaluar el sistema con grupos de estudiantes homogéneos, en cuanto a su nivel de conocimiento en Árboles de Decisión y de tamaño considerable. Por tal razón se está planificando una serie de experimentos que permitan validar la ventaja de utilizar el sistema SETECAD para enseñar Árboles de Decisión. Se propone utilizar SETECAD en la materia de Reconocimiento de Patrones cuyos grupos son alrededor de 30 estudiantes, en 4 grupos, y el procedimiento es dividir cada grupo en 2 partes, a una se le impartirá la materia sin permitirles utilizar SETECAD, y a los restantes que reciban como apoyo a su aprendizaje el sistema, y al final del tópico de Árboles de Decisión, realizar un análisis para determinar el nivel de aprovechamiento académico mostrado por ambos grupos que permita una evaluación experimental más realista de la utilización del sistema SETECAD.

5. Conclusiones

Las técnicas de clasificación supervisada son de gran ayuda para la solución de problemas de tipo predictivo. Los Árboles de Decisión son uno de los preferidos debido a su simplicidad y fácil entendimiento. Sin embargo, enseñar esta herramienta con técnicas de enseñanza tradicionales resulta complicado ya que son utilizadas diferentes operaciones matemáticas, las cuales se vuelven costosas cuando el número de atributos en un conjunto de entrenamiento a procesar es grande. En este trabajo se desarrolló un sistema computacional que permite generar Árboles de Decisión. Este sistema pretende mostrar el conocimiento de dicha técnica de enseñanza de una manera más dinámica, con la finalidad de comprender el por qué y de dónde sale cada resultado y al mismo tiempo darle al usuario la oportunidad de descubrir para que se utiliza. Además, SETECAD llevará de la mano a quien interactúe con él, para la comprensión de dicho algoritmo sin necesidad de realizar los cálculos manualmente. Como trabajo futuro se contempla la mejora del sistema propuesto, de acuerdo a los resultados obtenidos con los grupos de estudiantes de la materia de Reconocimiento de Patrones.

Agradecimientos. Los autores agradecen el apoyo brindado por PROMEP para el desarrollo de este trabajo con el recurso asignado en la carta de liberación PROMEP/103.5/12/8114, folio UAEH-PTC-554.

Referencias

1. Aybars Ugur, Ahmet Cumhur Kinaci. Web-Based Tool for Teaching Neuronal Network Concepts. Computer Applications in Engineering Education. Ed ISSUE 3. Volumen 18. PAG 449-457. 2010
2. Melvin Ayala, Malek Adjouadi, Mercedes Cabrerizo, Armando Barreto. A Windows-Based Interface for Teaching Image Processing. Computer Applications in Engineering Education. Ed ISSUE 2. Volumen 18. PAG 213- 224. 2010
3. Reyes Juárez-Ramírez, Guillermo Licea, Alfredo Cristo Bal-Salas. Teaching Undergraduate Students to Model use cases using Tree Diagram. Computer Applications in Engineering Education. Ed ISSUE 1. Volumen 18. PAG 77-86. 2010
4. Tom M. Mitchell. 1997. Machine Learning. Ed. Reviews. PAG 52
5. Tan, P.N., Steinbach, M. and Kumar, V. 2006. Introduction to Data Mining. Addison Wesley.
6. Data Mining Communitys Top Resource, <http://www.palisade.com/precisiontree/default.asp>
7. Data Mining Communitys Top Resource, http://http://www.gatree.com/?page_id=6
8. Data Mining Communitys Top Resource, <http://users.dsic.upv.es/~flip/smiles/>
9. Data Mining Communitys Top Resource, <http://www.rulequest.com/see5-info.html>
10. Data Mining Communitys Top Resource, <http://www.angoss.com/predictive-analytics-software/overview>
11. Weka - Data Mining with Open Source Machine Learning Software, <http://www.cs.waikato.ac.nz/ml/weka/>

Planificación de dietas con Answer Set Programming para pacientes con Diabetes Mellitus 2

Rosa Elena Pérez Ramírez¹, Claudia Zepeda Cortés¹
e Irma del Carmen Zamora Ginez²

¹Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla,
México

²Facultad de Medicina, Benemérita Universidad Autónoma de Puebla,
México

rosa.perez@solarium.cs.buap.mx, czepedac@gmail.com,
zamoraginezi@yahoo.es

Resumen La Diabetes Mellitus tipo II (DM2) es una enfermedad incurable y como parte del tratamiento de control los especialistas de la salud recomiendan una dieta personalizada, equilibrada y completa. La planificación de dietas requiere de procesar la información del paciente para obtener menús en los diferentes eventos alimenticios. En este trabajo se modela el problema de generar planes dietéticos para pacientes con DM2 basado en un enfoque declarativo como lo es *Answer Set Programming* (ASP).

Palabras clave: planificación, *Answer Set Programming*, representación del conocimiento, diabetes mellitus, planes dietéticos.

1. Introducción

La Inteligencia Artificial Médica se basa en la construcción de programas que ejecutan tipos de razonamiento como diagnóstico, planificación, toma de decisiones, etc. Algunas aplicaciones basadas en modelos simbólicos describen el problema con reglas y hechos para contestar preguntas sobre los objetos descritos y hallar soluciones a problemas particulares.

Por otro lado, tras el rezago de pacientes con DM2 con control metabólico positivo de la enfermedad, el Sistema de Salud Mexicano plasma un tratamiento para el control de la DM2 en la Norma Oficial Mexicana NOM-015-SSA2-2010 [1] con el apoyo del Sistema Mexicano Equivalentes de Alimentos (SMEA) [2].

La contribución de este trabajo es modelar el problema de agrupar un conjunto de alimentos que cumplan con ciertas restricciones para cada evento alimenticio (desayuno, comida, cena y dos colaciones), denominados en este trabajo planes dietéticos. La propuesta dirigida a pacientes con DM2 se basa en el SMEA y se desarrolla con un lenguaje de programación declarativo llamado ASP. ASP soluciona problemas de cálculo en términos de restricciones, excepciones y conocimiento incompleto y tiene aplicaciones tales como generar de planes de evacuación [3],

planificación de movimientos robóticos [4], planes de seguridad ante ataques [5], etc. La propuesta hace uso de DLV^k (implementación del paradigma de ASP) para modelar dicho problema.

El trabajo tiene 4 secciones. La sección 2 tiene una reseña de la operación de DLV^k en el modelado del problema de la planificación y bases de nutrición para generar planes dietéticos. En la sección 3 se desarrolla la propuesta: el modelo e implementación para generar planes dietéticos con DLV^k . Y en la sección 4 se presentan las conclusiones.

2. Marco teórico

Los fundamentos teóricos de este trabajo incluyen el funcionamiento de la herramienta DLV^k y el proceso detrás de generar planes dietéticos.

2.1. Planificación con DLV^k

DLV^{k1} basado en la semántica estable; los programas en DLV^k se conforman del Conocimiento de Fondo Estático del Dominio (CFED) y la Descripción del Dominio (DD). La DD se compone de fluentes, acciones y reglas que procesan la información del CFED, con la cual generará un plan. La estructura general de DD es como en la figura 1.

fluents: <declaración de fluentes >
actions: <declaración de acciones>
always: <reglas>
initially: <inicialización de reglas>
[noConcurrency.]
goal: <consulta>?(i)

Figura 1. Estructura de la descripción del dominio.

Los fluentes representan las propiedades básicas del sistema que pueden cambiar en el tiempo, las acciones representan las operaciones que permiten una transición de estados; la ejecución de una acción se preconditiona en el bloque iniciado con *executable* y causa efectos indicados en el bloque iniciado con *caused*. El recorrido a través de los estados arranca desde un estado inicial y se plantea llegar al estado final, a través de un plan que corresponde a una secuencia de acciones.

2.2. Cálculo de necesidades energéticas en la dieta

La dieta se forma por alimentos que aportan cantidades adecuadas de nutrientes, distribuidos en 5 eventos alimenticios al día. Los macronutrientes aportan energía (en kilocalorías, kcal) por cada gramo (g) del modo siguiente: 1 g de proteínas aporta 4 kcal, 1 g de lípidos aporta 9 kcal y 1 g de carbohidratos aporta 4 kcal [6].

¹ <http://www.dlvsystem.com/dlv/>

Para realizar el cálculo de necesidades energéticas del paciente de inicio se registran sus datos antropométricos y clínicos: nombre, edad, género, complexión, peso, estatura, niveles de glucosa registrados ya sea, por la prueba de Hemoglobina Glucosilada (HbA1c) ó la Prueba de Glucosa en Ayuno (PGA), y el nivel de actividad física determinado por el número de pasos promedio que el paciente camina al día. Luego se calcula el Índice de Masa Corporal (IMC) con la fórmula (1).

$$IMC = \frac{peso}{estatura^2} \quad (1)$$

Inmediatamente de acuerdo a [1] se analiza si el paciente tiene problemas de peso y en seguida se analiza el control de la enfermedad analizando los niveles de glucosa X según el tipo de la prueba realizada: si es la prueba es GPA $X \leq 126mg/dL$ indican control, y si la prueba es HbA1c $X \leq 7\%$ indican control.

Seguidamente de acuerdo a [7] se obtiene el peso ideal del paciente $[p_{min}, p_{max}]$, y se calcula el total de kcal recomendadas por día $[GET_{min}, GET_{max}]$ también llamado Gasto Energético Total (GET) con la fórmula (2), para ello se calcula el Gasto Energético en Reposo (GER) con la fórmula de Harris –Benedict (3) y (4) según el género del paciente y el Factor de Actividad (FA) física de acuerdo a [6].

$$GET = GER * FA \quad (2)$$

$$GER_H = 66.5 + (13.7 * peso) + (5 * estatura) - (6.8 * edad) \quad (3)$$

$$GER_M = 665.1 + (9.5 * peso) + (1.7 * estatura) - (4.7 * edad) \quad (4)$$

Las kcal recomendadas por día se distribuyen en los eventos alimenticios en proporciones mostradas en la tabla 1 del modo siguiente: para cada evento alimenticio E_i se calcula el intervalo de kcal recomendadas $[C_{min}, C_{max}]_i$ en la proporción indicada por los límites porcentuales $[e_{min}, e_{max}]_i$, con las fórmulas (5) y (6).

$$C_{min} = \frac{e_{min} * GET_{min}}{100} \quad (5)$$

$$C_{max} = \frac{e_{max} * GET_{max}}{100} \quad (6)$$

Tabla 1. Distribución de kilocalorías por ingesta diaria en cada evento alimenticio

Índice del Evento	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
Evento alimenticio E_i	Desayuno	Colación	Comida	Colación	Cena
Límites calóricos recomendados (%) $[e_{min}, e_{max}]_i$	25–30	10–15	25-35	10–15	15–20

Dado el estado de control de la enfermedad se establece la distribución de macronutrientes adecuada según la tabla 2 como sigue: se distribuye cada $[C_{min}, C_{max}]_i$ en cada macronutriente M_j se calcula el intervalo de kcal

recomendadas $[K_{min}, K_{max}]_{i,j}$ en la proporción indicada por los límites porcentuales $[m_{min}, m_{max}]_j$ con la fórmulas (7) y (8).

$$K_{min} = \frac{m_{min} * C_{min}}{100} \quad (7)$$

$$K_{max} = \frac{m_{max} * C_{max}}{100} \quad (8)$$

Tabla 2. Distribución de kilocalorías de macronutrientes en cada evento alimenticio

Índice de Macronutriente M_j	Macronutriente M_j	Límites recomendados para DM2 controlada (%) $[m_{min}, m_{max}]_j$	Límites recomendados para DM2 no controlada (%) $[m_{min}, m_{max}]_j$
$j=1$	Proteínas	13-15	15-20
$j=2$	Lípidos	25-35	35-45
$j=3$	Carbohidratos	50-60	40-50

Una vez obtenida una matriz de 5 x 3, donde cada elemento $[K_{min}, K_{max}]_{i,j}$ (límites de kcal recomendados para el evento E_i y el macronutriente M_j) en seguida se convierte a unidades en gramos $[g_{min}, g_{max}]_{i,j}$ con el uso de la fórmula (9) y la conversión de cada macronutriente (X) arriba especificados.

$$K_m / X = g_m \quad (9)$$

2.3. Uso del SMEA en los planes dietéticos

El SMEA contiene una base de datos de alimentos, donde cada alimento A_z tiene valores nutritivos y cantidades sugeridas, clasificados en 11 grupos denotado por S_k , cada elemento S_k con macronutrientes denotados por M'_{kj} , algunos se observan en la tabla 3. Un plan dietético denotado por P_i del evento alimenticio E_i se forma al agregar S_l elementos que en conjunto cumplen las restricciones $[g_{min}, g_{max}]_{i,j}$ y un máximo de repeticiones del elementos S_k en P_i denotado por n_k , luego se asocia cada elemento $S_l \in P_i$ a un elemento $A_z \in S_l$ de acuerdo a las preferencias del paciente.

Tabla 3. Extracto de grupos considerados por el SMEA

Elemento del SMEA S_k	Clasificación	Proteínas (g) M'_{k1}	Lípidos (g) M'_{k2}	Carbohidratos (g) M'_{k3}
S_1	Verduras	2	0	4
S_2	Frutas	0	0	15
S_3	Cereal sin grasa	2	0	15

Ejemplo. Sea un hombre adulto de 35 años, 70 kg, 170 cm, de complejión pequeña, 8% de la prueba HbA1c y sedentario. El análisis arroja que tiene IMC de 24.221, peso clasificado normal, su estado de DM2 es controlado y las necesidades

energéticas calóricas son $[GET_{min}, GET_{max}] = [2022, 2129]$. La distribución (g) en cada E_i y a su vez en cada M_j es como se muestra en la tabla 4.

Tabla 4. Resultado del cálculo de necesidades calóricas en un paciente con DM2

	Desayuno	Colación ₁	Comida	Colación ₂	Cena
Proteínas(g)	[15,28]	[6,12]	[15,28]	[6,12]	[9,20]
Lípidos(g)	[13,29]	[5,12]	[13,29]	[5,12]	[8,21]
Carbohidratos(g)	[70,112]	[28,48]	[70,112]	[28,48]	[42,80]

Un plan dietético P_2 para la primera colación podrían estar formado de 3 grupos S_i del modo siguiente: S_1 es 1 ración de fruta, S_2 es 1 ración de lácteo semidescremado, S_3 es 1 cereal con grasa, cumpliendo así con las restricciones de la tabla 4 para la primera colación. Luego se asocia cada S_i con los alimentos pertenecientes a dicho grupo y que sean de la preferencia del paciente, indicando las porciones recomendadas, como por ejemplo la siguiente asociación: 1 taza de fresa, 1 taza de yogurt semidescremado y $\frac{3}{4}$ de barra de granola.

En conclusión en la figura 2 se presenta el pseudocódigo que abstrae los pasos para generar planes dietéticos.

1. Inicio
2. Obtener_IMC()
3. Identificar_problemas_peso()
4. estado_control = Identificar_estado_control()
5. $[p_{min}, p_{max}] = \text{Buscar_peso_ideal}()$
6. $[GET_{min}, GET_{max}] = \text{Calcular_Getmax_y_Getmin}([p_{min}, p_{max}])$
7. $[m_{min}, m_{max}]_j = \text{Obtener_distribución_macronutrientes}(\text{estado_control})$
8. Para cada evento alimenticio E_i hacer
9. $[C_{min}, C_{max}]_i = \text{Obtener_porcentaje}([e_{min}, e_{max}]_i, [GET_{min}, GET_{max}])$
10. Para cada macronutriente M_j hacer
11. $[K_{min}, K_{max}]_{i,j} = \text{Obtener_porcentaje}([m_{min}, m_{max}]_i, [C_{min}, C_{max}])$
12. $[g_{min}, g_{max}]_{i,j} = \text{Convertir_kcal_gramos}([K_{min}, K_{max}]_{i,j})$
13. Fin_Para
14. $P_i = \text{Conjunto de } S_k = \text{Obtener_plan}([g_{min}, g_{max}]_{i,1}, [g_{min}, g_{max}]_{i,2}, [g_{min}, g_{max}]_{i,3})$
15. Conjuntos de $A_z = \text{Asociar_según_preferencias}(P_i)$
16. Fin_Para
17. Fin

Figura 2. Estructura de la descripción del dominio

3. Implementación del problema de planificación de dieta

En esta sección se trata el modelado para generar planes dietéticos mediante el uso de DLV^k tras haber calculado las restricciones. Para ello, y por cuestiones de espacio, se muestra la implementación de un ejemplo sencillo con datos e información simplificada que nos ayude a comprenderlo fácilmente. De la figura 2 en la línea 14 se describe la rutina llamada `Obtener_plan` que recibe como parámetros las

restricciones $[g_{min}, g_{max}]_{i,j}$ para cada evento alimenticio E_i , obteniendo como resultado un plan P_i , que es ejecutada por el entorno DLV^k .

Para realizar esta implementación, se supone se tiene un plato vacío al cual agregaremos elementos S_k y al realizar esta acción, también se incrementa el número de S_k agregados al plato y las cantidades de macronutrientes M'_{k1} (propiedades de S_k), ejecutando esta operación mientras se cumplen las restricciones de cada macronutriente $[g_{min}, g_{max}]_{i,j}$ y el máximo de posibles repeticiones del elemento S_k denotado por n_k , hasta tener un plato completo.

El CFED describe los elementos S_k que pueden conformar un plan dietético P_i para E_i , las restricciones $[g_{min}, g_{max}]_{i,j}$ y el valor máximo de veces que puede repetirse S_k en P_i como se muestra en la tabla 5.

Tabla 5. Elementos del Conocimiento del Fondo Estático del Dominio (CFED)

Descripción del elemento	Formato de la descripción	Ejemplo
Grupo S_k y valores respectivos de macronutrientes	<i>grupo(nombre_del_</i> S_k <i>,</i> <i>núm_gramos_proteínas_del_</i> S_k <i>,</i> <i>núm_gramos_lípidos_del_</i> S_k <i>,</i> <i>núm_gramos_carbohidratos</i> <i>del_</i> S_k <i>).</i>	De la tabla 3, elemento S_2 . <code>grupo(fruta, 0, 0, 5)</code> .
Límites de cada macronutriente para cada E_i	<i>x_macronutriente_lim(límite_infe</i> <i>rior, límite_superior).</i>	De la tabla 2, los límites de las proteínas. <code>proteinas_lim(3, 6)</code> .
Máximo de posibles repeticiones del elemento S_k en plan P_i	<i>rac_max(nombre_</i> S_k <i>,</i> <i>límite_superior).</i>	En un plan P_i como máximo se pueden agregar 5 verduras. <code>rac_max(verdura, 5)</code> .

3.1. Descripción del dominio (DD)

Este modelo considera los fluentes siguientes: la cantidad de cada macronutriente contenida en el plato, el número de cada S_k contenidos en el plato y el estado que describe si el plato está completo. Además la acción considerada es agregar un S_k al plato, instanciando los parámetros del elemento S_k , codificado como sigue:

```
actions:
  agregar(A, P, L, CH) requires grupo(A, P, L, CH).
```

Los efectos al ejecutar la acción agregar son el incremento de cada S_k que se agregan al plato y el incremento de los valores de cada macronutriente del S_k que se agrega. En el bloque de abajo se muestra el conteo de cada S_k que se agrega al plato tras ejecutar la acción agregar, proceso análogo para cada macronutriente.


```

always:
  caused enPlato(A,C_new,M)
  if C_new = C_old+1
  after agregar(A,P,L,CH), enPlato(A,C_old,M).
  caused -enPlato(A,C,M)
  after enPlato(A,C,M), agregar(A,P,L,H),
  rac_max(A,M), C <= M.

```

Un plato está completo si se han cumplido las restricciones de los macronutrientes $[g_{min}, g_{max}]_{i,j}$, instanciando variables en el bloque *after* y verificando los límites de cada variable en el bloque *caused*.

```

caused platoCompleto if TP = P+P1, TP >= MinP, TP <= MaxP,
TCH = CH+CH1, TCH <= MaxCH, TCH >= MinCH, TLip = Lip+Lip1,
  TLip >= MinLip, TLip <= MaxLip

after agregar(A,P1,Lip1,CH1),
  proteinas_lim(MinP,MaxP), proteinas_enPlato(P),
  lipidos_lim(MinLip,MaxLip), lipidos_enPlato(Lip),
  chidratos_lim(MinCH,MaxCH), chidratos_enPlato(CH).

```

La acción agregar se ejecuta mientras no se cumpla que el plato está completo y se cumplen las restricciones de raciones de cada S_k

```

executable agregar(A,P,L,CH)
if not platoCompleto, enPlato(A,R,M), R <= M.

```

El estado inicial describe las cantidades nulas de cada macronutriente presentes en el plato y la cantidad nula de S_k contenido en el plato. Además hace el uso de la negación como falla indicando que no tenemos evidencia de si el plato este completo hecho que mientras no sea verdadero se ejecuta la acción agregar como se muestra a continuación:

```
-platoCompleto.
```

3.2. Ejecución de la implementación

Se realiza la ejecución del código mediante el siguiente comando:

```
C:/dlv.exe genera_plan.plan evento_i.dl -FP -N=17
```

Obteniendo resultados en la tabla 6, donde tras una acción se cambia de estado, siendo el plan el conjunto de acciones realizadas para obtener un estado final.

Tabla 6. Ejecución de la implementación con DLV^k

Estado 0 ó estado inicial	Estado 1 Acción: agregar(lact sem,5,4,4)
-platoCompleto enPlato(fruta, 0, 5) enPlato(cereal_sinG, 0, 6) enPlato(azucar_conG, 0, 4) enPlato(lact_sem, 0, 1) chidratos_enPlato(0) lipidos_enPlato(0) proteinas_enPlato(0)	-platoCompleto enPlato(fruta, 0, 5) enPlato(cereal_sinG, 0, 6) enPlato(azucar_conG, 0, 4) enPlato(lact_sem, 1, 1) -enPlato(lact_sem, 0, 1) chidratos_enPlato(4) lipidos_enPlato(4) proteinas_enPlato(5) -chidratos_enPlato(0) -lipidos_enPlato(0) -proteinas_enPlato(0)
Estado 2 Acción: agregar(azucar conG,0,5,3)	Estado 3 Acción: agregar(cereal sinG,1,0,5)
-platoCompleto enPlato(fruta, 0, 5) enPlato(cereal_sinG, 0, 6) enPlato(azucar_conG, 1, 4) enPlato(lact_sem, 1, 1) -enPlato(azucar_conG, 0, 4) chidratos_enPlato(7) lipidos_enPlato(9) proteinas_enPlato(5) -chidratos_enPlato(4) -lipidos_enPlato(4)	-platoCompleto, enPlato(fruta, 0, 5) enPlato(cereal_sinG, 1, 6) enPlato(azucar_conG, 1, 4) enPlato(lact_sem, 1, 1) -enPlato(cereal_sinG, 0, 6) chidratos_enPlato(12) lipidos_enPlato(9) proteinas_enPlato(6), -chidratos_enPlato(7) -proteinas_enPlato(5)

Logrando un plan del modo siguiente:

```
agregar(lact_sem, 5, 4, 4);
agregar(azucar_conG, 0, 5, 3);
agregar(cereal_sinG, 1, 0, 5);
```

Indicando agregar un elemento de los siguientes grupos: S_1 es 1 lácteo semidescremado, S_2 es 1 ración azúcar con grasa, S_3 es cereal sin grasa que se asocia a alimentos dadas las preferencias del paciente.

4. Conclusiones y trabajo a futuro

El propósito de este trabajo es realizar una aplicación con fundamentos de la Inteligencia Artificial y Nutrición, por medio de la planificación con un lenguaje declarativo para obtener planes dietéticos para pacientes con DM2 que requieren un control y distribución adecuada de macronutrientes en la ingesta calórica. Dada la clasificación de grupos alimenticios del SMEA, este trabajo presenta como haciendo

uso de dicha clasificación se plante el problema de forma discreta y se resuelve con una implementación mediante descripción de reglas y hechos con DLV^k , obteniendo buenos resultados, además de ser una codificación relativamente fácil. Sin embargo para lograr una dieta se sugiere automatizar el pre-procesamiento de la información y analizar el estudio de preferencias del paciente, para presentar al paciente menús específicos de alimentos y no solo grupos, prototipo de un sistema en el cual actualmente se trabaja.

Referencias

1. Secretaría de Salud, Norma Oficial Mexicana NOM-015-SSA2-2010, Diario Oficial de la Federación. Noviembre 23 de 2010, Versión online: http://dof.gob.mx/nota_detalle.php?codigo=5168074&fecha=23/11/2010
2. Ana B. Pérez L., Berenice Palacio G., Ana L. Castro Becerra, Sistema Mexicano Equivalente de Alimentos, Fomento de nutrición y salud (2001)
3. Claudia Zepeda C. Evacuation Planning using Answer Set Programming. Tesis de Doctorado. Universidad de las Américas Puebla, 2005. Online <http://theses.insa-lyon.fr/publication/2005ISAL0117/these.pdf> (última fecha de verificación abril 2013).
4. Julián J. Portillo, Carmen L. García M., Pedro R. Márquez G. y Rogelio Baray A. Robot Platform Motion Planning using Answer Set Programming, Instituto Tecnológico de Chihuahua, México.
5. Chitta Baral, Michael Gelfond, Richard Scherl, Answer set programming as the basis for a Homeland Security QASk.
6. Daniel L. Román, Diego Bellido G., Pedro P. García L., Dietoterapia, nutrición clínica y metabolismo, 109 p. (2010)
7. José Mataix V., Nutrición y Alimentación Humana, Océano Ergon (2002)

Hacia la detección de impostores mediante un modelo basado en tareas

Jorge Rodríguez-Ruiz, Raúl Monroy y J. Benito Camiña-Prado

Tecnologías de Información y Computación
Tecnológico de Monterrey, Campus Estado de México
Atizapán, Estado de México
{a00965439, raulm, a00965049}@itesm.mx

Resumen La detección de impostores es un problema abierto; resolverlo tendrá un fuerte impacto, debido al riesgo inherente de si alguien tomara poder de nuestro computador. Tradicionalmente, este problema se ha estructurado como la detección de una anomalía en las acciones (comandos) realizadas por un usuario; sin embargo, los resultados de este enfoque no han sido satisfactorios. Recientemente, se ha propuesto detectar impostores identificando anomalías en el uso de los objetos (del sistema de archivos); es decir, no es la acción, sino sobre qué se realiza ésta lo que caracteriza fielmente a un usuario. En este artículo, se contribuye explorando esta idea un poco más; en particular, se considera que cada objeto puede asociarse con una tarea específica del usuario (por ejemplo, para un estudiante universitario, las tareas podrían ser: estudio, familia, etc.) El propósito final es establecer la siguiente hipótesis: mediante una secuencia relativamente corta de accesos a objetos podremos detectar la presencia de un impostor, pues traducida ésta en una secuencia de realizaciones de tareas se corrobore lo inusual que es con relación al perfil del usuario. En este artículo se detallan las bases del clasificador a construir, el tratamiento de las bitácoras de la base de impostores WUIL para extraer la información pertinente al modelo, y los experimentos que se realizarán para validar positiva o negativamente esta hipótesis.

Palabras clave: detección de intrusiones, conjunto de datos, grafos de navegación, seguridad computacional.

1. Introducción

Los ataques informáticos pueden originarse en el exterior o al interior de una organización (el hogar, una empresa, un cuerpo de gobierno, etc.) Los ataques internos son perpetrados dentro del perímetro seguro, por personas que disfrutan de nuestra confianza. Por ello, no es inusual que los atacantes internos tengan acceso directo a los recursos informáticos, mediante credenciales de acceso legítimas, y que, por lo tanto, sus acciones sean de hecho válidas con relación en la política implementada en equipos de seguridad. De lo anterior, se sigue, que es más difícil proteger a la organización de ataques internos. Dentro de la clase de ataques internos, existen los llamados de impostor, donde el perpetrador intenta

suplantar la identidad de un usuario legítimo. Estos ataques fácilmente pasan desapercibidos, pues el impostor tiene acceso a la máquina de la víctima objetivo, ya sea porque ésta abandonó una sesión sin cerrarla adecuadamente, o porque mediante la ejecución de algún código se obtuvieron las credenciales de acceso correspondientes.

El problema de detección de impostores se ha estructurado como uno de detección de anomalías. Ha sido estudiado activamente a partir del trabajo seminal de Schonlau *et al.* [18], que sugiere que para perfilar a un usuario es suficiente considerar el historial de comandos (sin parámetros) ejecutados por él durante una sesión UNIX. Para tal propósito, Schonlau *et al.* desarrollaron una base de impostores, comúnmente referida como *SEA* [17], y que ha sido aplicada como el estándar *de facto* para construir, validar y comparar una gran cantidad de métodos de detección (véanse, por ejemplo, [9,7]). Sin embargo, el desempeño de los detectores de impostores basados en SEA está lejos de ser satisfactorio [14]. Con el propósito de mejorar el poder de detección de impostores, se han considerado desde extensiones a SEA, incorporando, *e.g.*, argumentos de comandos [10], hasta el uso de información de actividad alternativa. Ejemplos de fuentes alternativas lo son el uso del teclado [6], u otros dispositivos E/S, como el ratón [4], o el uso de aplicaciones, *e.g.* un entorno gráfico para el manejo de documentos [16], o una abstracción que agrupa aplicaciones comunes con un tipo de actividad de bajo nivel (*e.g.*, la ejecución de cualquiera de los comandos `edit`, `vi`, o `xemacs` podría categorizarse como *edición*) [1].

En este artículo se adopta un enfoque alterno y novedoso [13,12], el cual construye el perfil de un usuario considerando los objetos del sistema de archivos que éste accede y cómo lo hace durante una sesión de trabajo cualquiera en su computador. Contrario a enfoques previos, este enfoque asevera que no son las acciones que realiza un usuario, sino el sobre qué las realiza el distinguido mayor que le caracteriza. Los usuarios provienen de los datos obtenidos en la base de datos WUIL [12], la cual se explica en una sección posterior.

Aquí, se considera que un usuario organiza y estructura su sistema de archivos de modo que es fácil distinguir las tareas que realiza en cada jornada. Así, hipotetizamos, el sistema de archivos de un usuario comprende un fólder por cada una de las múltiples tareas del mismo, y que comprenden todos los archivos con relación en dicha tarea. Note que en el modelo se abstrae el comportamiento de un usuario en términos de las tareas que realiza, pero que dichas tareas se elevan a un nivel superior, *e.g.* padre de familia, colaborador, finanzas, etc., y no como actividades de bajo nivel, *e.g.* edición, compilación, búsqueda, etc.

El clasificador es un proceso markoviano, adoptado de uno propuesto en [5]. Para cada usuario en WUIL, se construye su modelo, considerando una secuencia de tareas (obtenida tras asociar el objeto de cada registro en la bitácora del usuario a una tarea o rol de este último). El modelo perfila al usuario mediante un conjunto de rutas probabilísticas en la ejecución de tareas, y nos permite obtener, dada una secuencia de tareas en las que se alega la participación del usuario, una calificación sobre la disparidad de ésta conforme a su perfil. Entre

menos similitud haya entre la secuencia y el modelo mayor será la calificación de desemejanza.

Perspectiva general del artículo El resto de este artículo está organizado de la siguiente forma. Primero, en la sección 2, se muestran las limitaciones fundamentales de los métodos de detección de impostores, basados en SEA. Segundo, en la sección 3, se bosqueja la base de impostores WUIL, la cual se divide en bitácoras en condiciones normales del usuario y bitácoras que aleatoriamente incluyen ataques simulados. Tercero, en 4, se describirá cómo construir el proceso markoviano para la clasificación y posterior detección de impostores. Cuarto, en sección 5, se propone una serie de experimentos a realizarse a partir del conjunto de datos con el propósito de validar la hipótesis de que una secuencia relativamente corta de ejecuciones de tareas es suficiente para distinguir la presencia o no de un impostor. Finalmente, en sección 6, se reportan las conclusiones que hemos obtenido hasta el momento, y delineamos trabajo futuro.

2. Un Recuento del estado del arte en bases de impostores

El estudio de la detección de impostores ha resultado en un gran número de publicaciones, por lo que remitimos al lector a [2], o incluso uno menos reciente [15], si se desea tener un recuento de los trabajos más prominentes del área. Aquí nos enfocaremos sólo en discutir los conjuntos de datos que han sido colectados y desarrollados con el propósito de experimentar y comparar diferentes métodos de detección de impostores, así como en describir la hipótesis que motivó el desarrollo de los mismos.

2.1. SEA

SEA contiene bitácoras de 70 usuarios UNIX, cada uno con un registro total de 150,000 comandos, sin argumentos, colectados a través de la herramienta de auditoría `aact`. Schonlau *et al.* han dividido el registro de cada usuario en bloques, cada uno de tamaño 100 comandos, y llamado *sesión*. Así, cada usuario tiene asociadas 150 sesiones.

Los usuarios registrados en SEA son separados en dos clases: legítimos, 50 usuarios, e impostores, los 20 restantes. De cada registro de usuario legítimo, las primeras 50 sesiones no sufren modificación; éstas deben usarse en la construcción (entrenamiento) del método de detección de impostores; las últimas 100, sin embargo, son consideradas en la etapa de validación del método, e incluyen sesiones legítimas del usuario correspondiente, así como sesiones de uno o más de los usuarios impostores. SEA viene con una matriz que indica cuáles sesiones son legítimas, y cuáles impostor.

Si bien SEA hace posible realizar una comparación justa entre el desempeño de dos o más detectores de impostores, se ha mostrado que bajo condiciones más realistas de un ataque el desempeño de todos los detectores es verdaderamente pobre [14]. Adicionalmente, Schonlau *et al.* tomaron decisiones que afectan

dramáticamente la validez de los resultados. Por ejemplo, la selección de cuáles usuarios asumirían el rol de impostor no fue aleatoria, sino más bien arbitraria; más aún, algunos de estos supuestos impostores tienen un comportamiento inusualmente repetitivo, que puede detectarse incluso sin apoyo mecánico. Adicionalmente, todo impostor es, en principio, un usuario cualquiera interactuando ordinariamente con su computador, sin ninguna intención de usurpar la personalidad de un tercero o, peor aún, de lograr un objetivo (es decir, perpetrar un ataque). La decisión de si hay un impostor o no se toma al final de cada sesión, que contiene 100 comandos: una cantidad significativamente grande para que tenga sentido práctico.

2.2. Extensiones de SEA

De lo anterior, se desprende la necesidad por remediar dos problemas principalmente: uno, mejorar los desempeños de los detectores, y dos, mejorar las condiciones de validación de la detección, incluyendo condiciones de ataque más realistas. Curiosamente, la comunidad se ha enfocado principalmente en resolver el primer problema; trabajos como [8,10,9] sugieren mejoras metodológicas en el uso de SEA, así como la incorporación de opciones y argumentos en los comandos ejecutados.

Con relación en el segundo problema, hasta donde conocemos, existen sólo dos trabajos: uno, *RACoon* [3], un mecanismo que permite sintetizar sesiones tipo SEA con el propósito de mejorar el modelo de detección de cada usuario (pero que no ha sido apropiadamente validado), y otro, *SEA-I* [14], que incluye sesiones impostoras sintetizadas considerando una estrategia de ataque.

2.3. Otras bases de impostores

Recientemente, han habido algunos intentos por conformar bases de impostores que permitan primero realizar comparaciones justas en el desempeño de dos o más detectores, y, segundo, que consideren ambientes de ataque más realistas. El más prominente de todos es el trabajo de [6], quienes desarrollaron una base que considera la detección de impostores en la captura de una contraseña. Sólo en este enfoque, contrario a todos los predecesores y muchos de sus sucesores, tiene sentido el enfoque uno contra el resto, pues se considera que el impostor conoce la contraseña de la víctima objetivo. Este trabajo reporta resultados de comparar una gran cantidad de métodos previamente propuestos, usando una base común de información y uniformidad en el manejo de datos. Concluye que aún estamos lejos de lograr el estándar en la detección de impostores o autenticación de usuarios.

Un trabajo de interés en este tenor es el que reporta [1], aunque tiene sus be-moles irreparables. Presenta una base de impostores cuya hipótesis es que puedes separar a un usuario mediante su perfil de acciones (de bajo nivel), como edición, búsqueda, etc. Sin embargo, al momento de presentar resultados traicionan la hipótesis al considerar además número de archivos no tocados recientemente, etc. Otro defecto en este ejercicio es que es necesario categorizar manualmente cada

comando en una de las acciones arriba mencionadas. Pero el más devastador es que no incluye ataques realistas a las máquinas en las que fueron obtenidos los perfiles de usuarios, sino que se limitan a presentar una máquina similar a un gran número de usuarios y sobre ésta conducen un ejercicio de captura la bandera, dentro de un período de tiempo. El ejercicio del ataque es interesante, pero tendría que practicarse sobre la máquina de cada usuario, pues de otro modo carece de validez.

3. La base de impostores WUIL

Es necesario, para una correcta experimentación, contar con un conjunto de bitácoras confiable, tanto de usuarios, como de ataques. El conjunto de datos WUIL cuenta con un registro de datos de 20 usuarios y datos recopilados de tres ataques simulados directamente en las máquinas de dichos usuarios.

El perfil de los usuarios reclutados no es homogéneo, incluyendo estudiantes universitarios de pregrado y posgrado, personal administrativo y también de primera línea. El nivel de adecuación y experiencia en aspectos propios de sistemas operativos también varía de usuario a usuario, así como su capacidad de seguir buenas prácticas en el manejo de información. Afortunadamente, para todos ellos WUIL incluye información moderadamente detallada sobre todos estos y otros aspectos que pueden en un momento dado usarse para explicar forasteros (*outliers*) en los resultados de detección de impostores obtenidos.

Para el diseño y posterior simulación de ataques impostores, los creadores de WUIL recurrieron a una encuesta, la cual contestaron casi 50 personas. Consensando los resultados de dicho ejercicio, diseñaron tres tipos de ataques: básico, intermedio y avanzado, los cuales consideran, respectivamente, un atacante ocasional sin ningún objetivo claro por comprometer al usuario víctima, otro que, por el contrario, tiene esa intención, pero que no posee herramientas para perpetrarlo, y otro atacante con más intención, dotado de herramientas, *scripts* entre otros. Estos tres ataques fueron simulados de manera uniforme y controlada, directamente en la máquina de los usuarios participantes, recopilando bitácoras y destruyendo toda información que pudiera comprometerles.

4. Creación del modelo de representación del usuario basado en tareas

4.1. Definición de Tarea

Para clasificar el comportamiento de cada usuario se usará el concepto de tareas, el cual se define como una agrupación lógica de carpetas con sus respectivos archivos y subcarpetas, los cuales utiliza el usuario para un fin en específico.

4.2. Obtención de las tareas a partir del conjunto de datos

Una vez definido el concepto de tareas es necesario saber cómo obtenerlas a partir de la información contenida en WUIL. El mapeo de carpetas y archivos

a tareas tiene cierta complejidad, ya que depende del orden que mantenga el usuario, la correcta separación de las carpetas asociadas a cada tarea y la granularidad con la que el usuario consideraría qué es una tarea y qué no lo es. La granularidad es un problema ya que si es muy gruesa cada archivo pasa a ser una tarea, lo que eliminaría la necesidad de agrupamiento, mientras que si es muy fina se pierden ciertos elementos que podrían indicar un mejor patrón de comportamiento del usuario. Para resolver el problema de la granularidad y evitar depender de la correcta separación en tareas por parte del usuario se decidió que se realizaría una asignación automática de las tareas a partir de los datos de la profundidad de navegación en el sistema de archivos contenidos en los datos de cada usuario. Se calculará entonces un punto de corte P por cada usuario el cual servirá para indicar en qué profundidad de navegación se mapea una carpeta como a una tarea, mientras que todos los archivos y subcarpetas contenidos en ésta pertenecerían a la misma tarea. Dentro de nuestros análisis preliminares pudimos observar que existen diferencias marcadas entre los patrones de comportamiento con respecto a la profundidad de navegación de un usuario normal y un atacante. En la Figura 1 podemos observar cómo estas variaciones son claras.

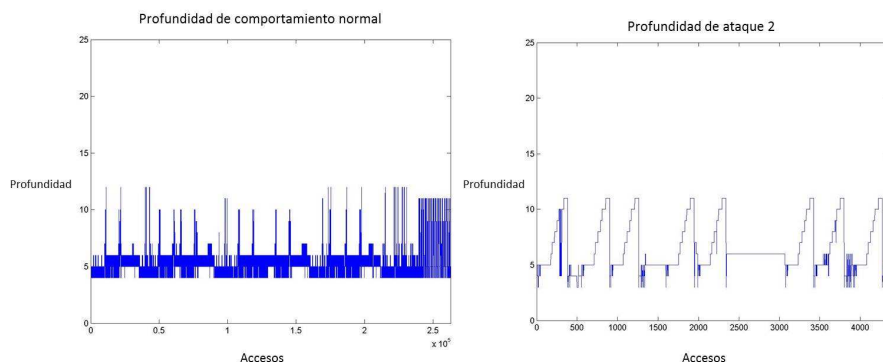


Figure 1. Comparación de comportamiento de profundidad de navegación del usuario contra profundidad de navegación de los ataques.

Al saber cuáles son las tareas realizadas por un usuario, se procede a realizar un mapeo donde a cada tarea se le asigna un valor numérico que la representará, teniendo así que el alfabeto a utilizar para la construcción del modelo y la utilización del clasificador contiene solamente elementos numéricos. La ventaja de tener solamente elementos numéricos es que permite un procesamiento más rápido al no tener que parsear por cada vez que se tenga una tarea todo el string que compone el nombre de la misma. El alfabeto a usar no es de tamaño fijo, ya que es del mismo tamaño que el número de tareas que un usuario genere y puede ser aumentado al momento de realizar una actualización del modelo para tomar en cuenta tareas que en un inicio existían. Para el caso de los archivos y carpetas

que no pertenezcan a ninguna tarea se les asignara un valor numérico temporal, con el cual serán identificadas al momento de realizar una clasificación. Este valor se puede hacer permanente al realizar una actualización del modelo.

4.3. Construcción del modelo

Una vez que se tienen las secuencias de tareas es necesario crear un modelo que nos permita obtener el comportamiento normal de un usuario. Para realizar este modelo se utilizó el método descrito en [5], en el cual se utiliza una cadena de Markov para crear la representación del comportamiento y posteriormente usarlo como un clasificador de una sola clase. La selección fue dada por la discusión en [11] dado que las cadenas de Markov permiten trabajar con datos no numéricos y tomar en cuenta un factor temporal, elemento que es importante para nosotros puesto que necesitamos observar cómo progresa a partir del tiempo y dar una predicción que nos permita decir esto. También se puede a partir de este modelo realizar la construcción de un clasificador de una sola clase, el cual puede detectar anomalías a partir de solamente los datos del usuario sin tener que tomar en cuenta las bitácoras de ataques para lograr el entrenamiento.

Se necesitan definir algunos elementos de soporte que nos permitirán entender mejor cómo funciona el algoritmo que permite realizar la construcción del modelo, entre ellos una estructura que servirá para el cálculo de las probabilidades, los parámetros de control y algunas funciones que se realizan dentro del algoritmo. El primer elemento a definir es un estado e , el cual es una secuencia de tareas desde o hacia la cual se tienen transiciones. Para saber la cantidad de tareas que contiene cada estado se utiliza un parámetro de ventana w , el cual nos sirve para realizar un agrupamiento de una serie de tareas en un sólo estado, permitiendo representar de esta forma mejor las tareas que se realizan juntas en un traza. Se tiene también una estructura E que contiene todos los estados, así como sus transiciones y las veces que se ha observado dentro de los datos que se realizó esta transición. Por último se tiene una función corrimiento que nos permite dada una tarea y un estado, tomar la secuencia de tareas contenidas en dicho estado y crear un nuevo estado donde la tarea más vieja de esa secuencia, la cual se encuentra más a la izquierda dentro de la secuencia, se elimina en el nuevo estado y posteriormente se agrega la nueva tarea, lo que permite preservar el tamaño de la secuencia en concordancia con el tamaño de ventana w .

Antes de realizar los pasos del algoritmo se necesita tener una inicialización por cada traza que se incluya para crear el modelo, donde se tendrán dos estados llamados *actual* y *siguiente*. Cada estado esta etiquetado con una secuencia de tareas de tamaño w y el estado inicial por ejemplo tendría una etiqueta nulo,nulo si $w=2$. Este estado nulo servirá como estado ficticio de inicio, lo que nos permite modelar un usuario que no siempre inicie sus actividades con la misma tarea. Si es la primera traza que se va a usar se agrega el estado que contiene solamente tareas nulas a la estructura E . Después de esta inicialización se realiza una iteración de los siguientes pasos por cada tarea que exista en la traza.

1. Se realiza la función corrimiento con el estado *siguiente* y la nueva tarea leída de la traza.

2. Se verifica si existe una transición en E desde el estado *actual* hacia *siguiente*. Si la transición existe se suma 1 al contador asociado a las veces que esa transición se ha visto, en caso contrario se agrega la transición y el contador se inicializa en 1.
3. Se convierte el estado *siguiente* en el estado *actual*.
4. Si el estado *actual* no está contenido en E , se agrega.

Una vez que se han analizado todas las trazas se procede a realizar por cada estado en la estructura E el cálculo de probabilidad que a partir de un estado e se avance a un estado e' para todas las transiciones que existan desde e . Recordemos que si no existe una transición significa que ese comportamiento no se observó y por lo tanto no se incluye en el modelo. Para calcular la probabilidad de una transición entre e y e' se utiliza la siguiente formula:

$$\Pr(e, e') = \frac{\text{Número de transiciones observadas}(e, e')}{\sum_{i=0}^n \text{Número de transiciones observadas}(e, e'_i)}$$

Si tomamos que n es el número de estados a los cuales existe una transición desde el estado e , entonces tenemos que la probabilidad de que exista una transición es la proporción de la frecuencia de una transición entre e y e' y la cantidad de transiciones que parten desde e .

4.4. Actualización del modelo

Dentro de los sistemas de cómputo la estructura de los archivos no tiene un estado estable. Es por esto que para que el modelo pueda representar de una forma fiel el comportamiento de un usuario es necesario estarlo actualizando con las nuevas tareas que se vayan creando conforme el paso del tiempo. La estructura que se manejó anteriormente para la creación del modelo es la clave para realizar esto, ya que al estar guardada simplemente se tiene que actualizar tanto los contadores de transiciones ya existentes, los cuales refuerzan ciertos comportamientos en el modelo, así como agregar los nuevos estados y transiciones, para que estos no sean considerados anómalos en un futuro.

4.5. Utilización del modelo en el clasificador

Para poder clasificar una secuencia de tareas a partir del modelo creado se tiene que aplicar el mismo tratamiento que a los datos originales, es decir obtener las tareas a partir de una traza y realizar el mapeo numérico que represente cada tarea. A partir de aquí se utiliza el algoritmo detallado en [5] para lograr una clasificación a partir del modelo generado en el paso anterior. Para que el algoritmo pueda funcionar se toma en cuenta que existe por cada traza a analizar los valores X y Y que se irán actualizando conforme se vayan observando las tareas y comparándolas con el modelo, un valor Z entre 0 y 1 que servirá para penalizar el no encontrar una secuencia de tareas. También es necesario que

dentro del modelo se cumpla que para un estado e la suma de las probabilidades de las transiciones de e a todos los estados e' sucesores de e sea equivalente a 1, es decir que la función de probabilidad quede como sigue:

$$\sum_{e' \in \text{sucesores}(e)} \Pr(e, e') = 1$$

En [5] se puede observar que dependiendo del método de actualización de los valores X y Y es la efectividad del clasificador que se tiene, además de que para no tener diferentes algoritmos se manejan funciones vacías que sirven de comodín ya que después son reemplazadas por la función correcta de actualización. De los métodos propuestos seleccionamos el que nos da la probabilidad de falla, es decir el que nos indica la probabilidad de que una traza no exista o sea considerada anómala por tener una baja probabilidad de ocurrir. Al tener una selección no es necesario usar las funciones comodín que se manejan y se puede definir directamente el algoritmo de actualización de los valores X y Y . Para cada traza a analizar se considera que los valores de los coeficientes X y Y son inicializados en 0, además de que se tienen dos estados: *actual* y *siguiente*, los cuales son etiquetados con una serie de tamaño w de tareas nulas. Después de esta inicialización se realiza una iteración de los siguientes pasos por cada tarea que exista en la traza.

1. Se realiza la función corrimiento con el estado *siguiente* y la nueva tarea leída de la traza. En cuanto se han leído todas las tareas de una traza se termina de iterar.
2. Se actualizan los coeficientes de la siguiente manera.
 - Caso 1: Existe una transición en el modelo desde el estado *actual* hacia el estado *siguiente*. Si ocurre este caso se actualizan los coeficientes X y Y de la siguiente forma:

$$\begin{aligned} X &= X + 1 \\ Y &= Y + \sum_{e \in \text{sucesores}(\text{actual}) \wedge \text{siguiente} \neq e} P(\text{actual}, e) \end{aligned}$$

- Caso 2: No existe una transición en el modelo desde el estado *actual* hacia el estado *siguiente*. Si ocurre este caso se actualizan los coeficientes X y Y de la siguiente forma:

$$\begin{aligned} X &= X + 1 \\ Y &= Y + Z \end{aligned}$$

donde Z es la penalización que se da por no encontrar la transición (*actual*, *siguiente*) en el modelo. Es importante notar que el valor de Z sera estimado empíricamente durante los experimentos.

3. Se convierte el estado *siguiente* en el estado *actual*.

Una vez que se ha terminado de analizar la traza, se calcula el valor $\mu(\text{traza}) = Y/X$, que indica qué tan bien predice la cadena de Markov la traza. Un $\mu(\text{traza})$ bajo nos indica que la cadena tiene mayores probabilidades de ser precedida por el modelo. Para terminar, se tiene que dar una clasificación de si la traza es normal o anómala, por lo que se usa un valor de umbral r , donde:

$$\text{Clasificación}(\text{traza}) = \begin{cases} \text{Anómalo} : \mu(\text{traza}) \geq r \\ \text{Ordinario} : \text{En caso contrario} \end{cases}$$

5. Experimentos propuestos

Recordemos que la hipótesis es mostrar que con una secuencia relativamente corta de tareas podemos detectar un impostor, para lo cual se usará el modelo y clasificador descritos anteriormente. Para lograrlo, se tienen que resolver dos problemas principales: el encontrar el punto de corte P para determinar la profundidad desde la cual se tomarán las tareas y el ajuste de los parámetros del clasificador para que se pueda maximizar la detección de anomalías reduciendo la cantidad de falsos positivos. Para la determinación del punto de corte se necesita encontrar un método automático que pueda realizar esta acción. Se propone iniciar con un análisis estadístico a profundidad donde los datos de cada usuario se tomarán como una muestra y realizar un análisis para determinar cómo están distribuidas las profundidades de los usuarios. Creemos que a partir de estos análisis podemos encontrar un punto de corte que preserve la información necesaria para la correcta distinción entre un impostor y un usuario legítimo, permitiendo también la abstracción de los comportamientos en una serie de tareas. Una vez encontrado cómo se distribuyen los datos podemos determinar una combinación de estos análisis estadísticos que arroje un valor que se pueda usar para la determinación de las tareas.

El siguiente punto a tratar es cómo se realizarán los experimentos para observar el funcionamiento del clasificador, sin embargo es necesario determinar cómo se realizará el ajuste de parámetros, ya que actualmente no se tiene un valor establecido para el tamaño de ventana w , la penalización z y el umbral r . Estos parámetros se tendrán que ir probando conforme a los experimentos para determinar el comportamiento del clasificador con ligeros cambios en ellos. Proponemos fijar valores basándonos en [5], y a partir de ahí tomar un valor para realizar experimentos con él, para después aplicar análisis estadísticos a los resultados y determinar los mejores valores para los parámetros. Para validar que nuestro clasificador basado en un modelo a partir de las tareas funciona correctamente se tomará entre un 15 % y 25 % de las trazas, lo cual evitará que colaboren a la creación del modelo de comportamiento normal, para ser usadas en un esquema de validación cruzada. Al ser un clasificador de una sola clase no necesitamos entrenarlo con las trazas de ataques, ya que esto sería contraproducente puesto que sólo podría clasificar correctamente algunos ataques que ya se le hubieran mostrado anteriormente. Por lo que se usarán las trazas que se excluyeron de la creación del modelo y las trazas de los ataques para validar que el modelo construido para cada usuario es representativo de su comportamiento

y por lo tanto las trazas que pertenecen a comportamientos legítimos son clasificadas como legítimas, mientras que las trazas que pertenecen a un ataque son clasificadas con anómalas. Es importante recalcar que las trazas son de tamaños variables, por lo que si se logra obtener con trazas relativamente pequeñas una correcta clasificación sin incurrir en muchos falsos positivos o falsos negativos, entonces habremos validado nuestra hipótesis.

6. Conclusiones y trabajo futuro

En este trabajo se tomó un enfoque diferente al ya existente en la literatura para la detección de impostores, ya que estamos utilizando las acciones que hace el usuario sobre su sistema de archivos para obtener las tareas que realiza y así obtener un modelo de su comportamiento normal. Para lograr la clasificación se buscó un clasificador de una sola clase y basados en la discusión de [11], se optó por un clasificador basado en cadenas de Markov, método que se tomó de [5] y se adaptó para que funcionara en base a nuestro concepto de tareas. Actualmente se encuentran desarrollados los mecanismos que permiten la creación de un modelo a partir de trazas de tareas y el mecanismo que permite realizar una clasificación basada en el modelo. Con estos mecanismos desarrollados es posible empezar a realizar los experimentos propuestos. Análisis preliminares muestran que utilizando una clasificación basada en tareas se puede lograr esto, análisis como el de profundidades, que al mostrar gráficamente las profundidades de navegación podemos observar cómo emergen patrones los cuales son diferentes a los que se muestran en las trazas de los atacantes. Los resultados esperados a partir de los experimentos definidos, es llegar a demostrar que con nuestro enfoque se puede tener a partir de una serie relativamente corta de acciones, una clasificación que permita saber cuándo en el sistema está actuando un usuario legítimo o un impostor. A mediano plazo se planea la comparación contra otros clasificadores basándose en los mismos datos, puesto que esto nos puede mostrar deficiencias en algunos clasificadores, así como mejores formas de realizar la detección de impostores.

Referencias

1. Ben-Salem, M., S., S.: Modeling user search behavior for masquerade detection. In: Sommer, R., Balzarotti, D., Maier, G. (eds.) Proceedings of the 2011 International Symposium Research Advances in Intrusion Detection, RAID'11. Lecture Notes in Computer Science, vol. 6961, pp. 181–200. Springer (2011)
2. Bertacchini, M., Fierens, P.: A survey on masquerader detection approaches. In: Proceedings of V Congreso Iberoamericano de Seguridad Informática. pp. 46–60. Universidad de la República de Uruguay (2008)
3. Chinchani, R., Muthukrishnan, A., Chandrasekaran, M., Upadhyaya, S.: RACOON: Rapidly generating user command data for anomaly detection from customizable templates. In: Proceedings of the 20th Annual Computer Security Applications Conference, ACSAC'04. pp. 189–204. IEEE Computer Society Press (2004)

4. Imsand, E.S., Hamilton, J.A.J.: Masquerade detection through GUID. In: Proceedings of the Global Communications Conference, 2008. GLOBECOM'08. pp. 2104–2108. IEEE Computer Society Press (2008)
5. Jha, S., Tan, K.M.C., Maxion, R.A.: Markov chains, classifiers, and intrusion detection. In: Proceedings of the 14th IEEE Computer Security Foundations Workshop, CSFW'01. pp. 206–219. IEEE Computer Society Press (2001)
6. Killourhy, K.S., Maxion, R.A.: Why did my detector do *that?!* - predicting keystroke-dynamics error rates. In: Jha, S., Sommer, R., Kreibich, C. (eds.) Recent Advances in Intrusion Detection, 13th International Symposium, RAID 2010. Lecture Notes in Computer Science, vol. 6307, pp. 256–276. Springer (2010)
7. Latendresse, M.: Masquerade detection via customized grammars. In: Julish, K., Kruegel, C. (eds.) Proceedings of the Second International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA 2005. Lecture Notes in Computer Science, vol. 3548, pp. 141–159. Springer (2005)
8. Maxion, R.A., Townsend, T.N.: Masquerade detection using truncated command lines. In: Proceedings of the International Conference on Dependable Systems & Networks. pp. 219–228. IEEE Computer Society Press, Washington, DC (June 2002)
9. Maxion, R.A., Townsend, T.N.: Masquerade detection augmented with error analysis. IEEE Transactions on Reliability 53(1), 124–147 (2004)
10. Maxion, R.A.: Masquerade detection using enriched command lines. In: Proceedings of the International Conference on Dependable Systems and Networks, DSN'03. pp. 5–14. IEEE Computer Society Press, San Francisco, CA, USA (June 2003)
11. Mazhelis, O.: One-class classifiers : a review and analysis of suitability in the context of mobile-masquerader detection. South African Computer Journal 36, 29–48 (2006)
12. Cami na Prado, J.B., Monroy, R., Trejo, L.A.: The windows-users and -intruder simulations logs dataset (wuil): An experimental framework for masquerade detection mechanisms. Expert Systems with Applications ?, ?-? (2013), submitted for publication
13. Camiña Prado, J.B., Monroy, R., Trejo, L., Sánchez, E.: Towards building a masquerade detection method based on user file system navigation. In: Batyrshin, I.Z., Sidorov, G. (eds.) Proceedings of the 10th Mexican International Conference on Artificial Intelligence, MICAI'11. Lecture Notes in Artificial Intelligence, vol. 7094, pp. 174–186. Springer (2011)
14. Razo-Zapata, I., Mex-Perera, C., Monroy, R.: Masquerade attacks based on user's profile. Journal of Systems and Software 85(11), 2640–2651 (2012)
15. Salem, M.B., Hershkop, S., Stolfo, S.J.: A survey of insider attack detection research. In: Stolfo, S.J., Bellovin, S.M., Hershkop, S., Keromytis, A., Sinclair, S., Smith, S.W. (eds.) Insider Attack and Cyber Security: Beyond the Hacker, pp. 69–90. Advances in Information Security, Springer (2008)
16. Sankaranarayanan, V., Pramanik, S., Upadhyaya, S.: Detecting masquerading users in a document management system. In: Proceedings of the IEEE International Conference on Communications, ICC'06. vol. 5, pp. 2296–2301. IEEE Computer Society Press (2006)
17. Schonlau, M.: Masquerading user data (2008), <http://www.schonlau.net>
18. Schonlau, M., DuMouchel, W., Ju, W., Karr, A., Theus, M., Vardi, Y.: Computer intrusion: Detecting masquerades. Statistical Science 16(1), 58–74 (2001)

WUIL: Una base de datos para probar mecanismos de detección de intrusos

J. Benito Camiña y Raúl Monroy

Departamento de Ciencias Computacionales,
Campus Estado de México del Tecnológico de Monterrey,
Atizapán, Estado de México, México
{a00965049,raulm}@itesm.mx

Resumen En el mundo actual, donde la información tiene un valor enorme, se ha vuelto una parte vital la detección de intrusos durante una sesión en la computadora. El estudio de la detección de intrusos tomó fuerza a partir del trabajo de Schonlau *et al.* [1], en el cual presentan una base de datos para comparar diversos métodos de detección de intrusos con base en comandos en UNIX. Sin embargo, la base de datos y el método en sí de detectar intrusos con base en comandos no fue lo suficientemente útil. Por tal motivo se empezaron a realizar nuevas bases de datos basándose en otras fuentes de comportamiento del usuario, pero aún no se han encontrado los resultados esperados. En este artículo se presenta una base de datos que pretende cubrir algunas de las deficiencias de bases de datos pasadas y que se base en el comportamiento del usuario en su interacción con el sistema de archivos de su computadora. Además se presentan métodos de cómo utilizaremos la base de datos para detectar intrusos.

Palabras clave: base de datos, detección de intrusos, seguridad computacional.

1. Introducción

Debido a la cantidad de información sensible que se guarda en las computadoras, el problema de detectar intrusos de forma rápida se ha vuelto algo primordial en los últimos años. Esto con el fin de evitar que las personas tengan pérdidas incalculables.

La detección de intrusos es regularmente abordada como una tarea de detección de anomalías, donde se crea el perfil del usuario con base en su comportamiento y si el nuevo comportamiento se desvía del comportamiento normal, se detecta como un intruso. Esto se comenzó a estudiar principalmente a partir del trabajo de Schonlau *et al.* [1] donde perfila a los usuarios con base en los comandos que utiliza durante una sesión en UNIX. Schonlau *et al.*'s crearon una base de datos, llamada SEA, la cual se utilizó durante mucho tiempo como el estándar para desarrollar y comparar mecanismos de detección de intrusiones.

Lamentablemente SEA presenta limitaciones. La más importante es que no cuenta con ataques reales, en lugar de eso utiliza el método de Uno-Contra-Los-Otros (OVTO, por sus siglas en inglés). Utilizando este método, una sesión de un usuario se compara contra alguna de otro usuario y se busca detectar diferencias en su comportamiento, sin que en realidad haya un ataque. Además, se ha probado que el uso de comandos para distinguir a un usuario de un atacante no es lo suficientemente poderoso [2].

Como resultado las investigaciones se han centrado a otros tipos de fuente de actividades del usuario, como el uso de dispositivos, principalmente el teclado [3,4]. Con esto se han creado nuevas bases de datos, sin embargo, siguen con el problema de utilizar el método OVTO.

Por lo tanto se decidió crear la base de datos: Bitácoras de Usuarios e Intrusos simulados para Windows (WUIL, por sus siglas en inglés). Los registros de WUIL contienen información del qué y el cómo accede los archivos un usuario. Con esto es posible crear un perfil del usuario y determinar si un comportamiento nuevo pertenece o no al usuario.

En este trabajo se maneja la hipótesis de que hay ciertos rasgos en la forma en que un usuario interactúa con el sistema de archivos, con los cuales se puede diferenciar de un intruso. Además, se muestran varias formas de validar esta hipótesis utilizando los siguientes conceptos: localidad, tareas, número de accesos y de accesos nuevos.

La base de datos WUIL está disponible en:

<http://homepage.cem.itesm.mx/raulm/wuil-ds/>

La organización del documento es la siguiente, primero en la sección 2 se muestra el trabajo relacionado, donde se habla de SEA y otras bases de datos que se han realizado. En la sección 3 se presenta la base de datos WUIL y se explica cómo se realizó esta base de datos. Luego en la sección 4, se muestra el trabajo en proceso que tenemos y cómo vamos a detectar intrusos utilizando la base de datos. Finalmente, en la sección 5 se presentan las conclusiones y el trabajo futuro.

2. Trabajo relacionado

En los últimos años las investigaciones en el área de detección de intrusiones ha sido abundante. Los trabajos realizados se han basado en el análisis de diferentes fuentes de actividad del usuario. Sin embargo, los trabajos realizados no han brindado resultados concluyentes y se continúan realizando investigaciones en el área.

2.1. La Base de datos SEA

A partir del trabajo realizado por Schonlau *et al.* [1] en el cual presentan a SEA [5] el interés en el área de detección de intrusiones ha ido en aumento. SEA es una base de datos que permite probar distintos mecanismos de detección de

intrusiones. La base de datos contiene bitácoras de la actividad de 70 usuarios en el sistema operativo UNIX, cada una de las bitácoras de los usuarios consiste de 15,000 comandos sin argumentos.

El mayor problema que tiene SEA es que utilizan el concepto de OVTO en el cual no se tienen ataques realizados en las máquinas de los usuarios, si no que utiliza bitácoras de ciertos usuarios como usuarios legítimos y bitácoras de otros usuarios como bitácoras contaminadas. Para lograr esto primero seleccionaron 50 usuarios como usuarios legítimos y los otros 20 los consideran, de forma artificial, como intrusos. Después, dividen las bitácoras de los usuarios en bloques de 100 comandos, cada bloque recibe el nombre de sesión, con lo cual quedan 150 sesiones por usuario. Las primeras 50 sesiones no se modifican y se utilizan para construir el perfil del usuario, las otras 100 sesiones se utilizan para la validación de la base de datos y pueden o no estar contaminadas.

Cada sesión de validación puede estar o no totalmente contaminada, si está contaminada se dice que hay un intento de intrusión. Para crear una sesión contaminada lo que realizan es el reemplazo de una sesión de validación por una de un usuario seleccionado como intruso de manera arbitraria. Ellos brindan una matriz donde indican por cada usuario cuáles son las sesiones que están contaminadas.

Como se puede ver SEA brinda la posibilidad de comparar distintos métodos de detección de intrusos, sin embargo la información que brindan es artificial al no haber ataques reales. Además, de que el únicamente considerar los comandos para detectar intrusos, es insuficiente [2]. Por tal motivo los trabajos en el área de detección de intrusiones han buscado nuevas fuentes de actividad para detectar intrusos como son el uso del algún dispositivo, como el teclado [3,4] o el comportamiento abstracto del usuario [6].

2.2. Otras bases de datos

Una de las fuentes de actividad alternativa que más se ha estudiado es el análisis de los patrones de tecleo [7,8,9,10,11,12,13]. En la mayoría de estos trabajos se utiliza el tecleo de una contraseña, la cual es repetida cierto número de veces. Es muy difícil comparar estos trabajos, ya que cada uno utiliza su propia base de datos y varían aspectos entre ellas como son la información que guarda cada uno o el número de veces que repiten la contraseña.

Uno de los trabajos más relevantes en cuanto al análisis de los patrones de tecleo es el de Maxion y Killourhy [4]. En este trabajo utilizan 51 usuarios para recolectar 400 registros por cada uno de ellos, en varias sesiones de 50 registros en diferentes días. Ellos recolectan información que va desde el tiempo en el que presionan las teclas, hasta el género o si el usuario es zurdo o diestro para tener una base de datos completa.

Aunque guardan información muy completa de los usuarios, su propuesta presenta dos problemas principales. Primero, porque utilizan de nuevo el enfoque OVTO, ya que todos los usuarios escriben la misma contraseña y lo que buscan es diferenciar como escribe un usuario la contraseña contra los demás, aunque es razonable porque el trabajo es sobre el conocimiento y forma de capturar

una contraseña y si fueran diferentes sería complejo poder realizar el estudio. Segundo el tener que escribir 50 veces seguidas por sesión la misma contraseña, es poco realista.

Por otro lado, hay trabajos que consideran combinaciones de distintas fuentes de actividad como es el caso de [6] en el cual utilizan 22 tipos de actividades del usuario, incluyendo qué archivos del sistema se utilizan, comunicaciones, búsquedas, etc. La base de datos se utiliza para modelar el comportamiento a la hora de realizar búsquedas principalmente. Cuentan con 18 usuarios y con intrusiones simuladas de 40 diferentes individuos, durante un período de cuatro días y tienen más de 500,000 registros por usuario en promedio. El problema es que son muy pocos días de recolección de información y aun así el tamaño del conjunto de datos es de diez Gbytes.

En ese trabajo se les ordenó a los intrusos actuar siguiendo uno de tres distintos escenarios: *malicioso*: el objetivo era encontrar información financiera de un compañero de trabajo en un período de 15 minutos; *benigno*: similar al primero pero el intruso utiliza la computadora del compañero para propósitos buenos, asumiendo que no tiene acceso a su computadora por alguna razón; *neutral*: los intrusos tenían la libertad de ver lo que quisieran en la computadora del compañero de trabajo. Un problema es que todo lo que manejan depende del Sistema Operativo utilizado.

Un trabajo muy similar al que se presenta en este documento fue realizado con anterioridad [14] y los resultados fueron alentadores. En ese trabajo también se analizaba la interacción del usuario con el sistema de archivos. Sin embargo, sólo se utilizaban tres usuarios y son muy pocos para dar resultados concluyentes.

2.3. Desventajas de las bases de datos actuales

En este punto pretendemos mostrar a manera de resumen los tres problemas primordiales que creemos que existen en las bases de datos actuales:

1. Varían en cuanto el número de usuarios, tiempo de captura y plataforma utilizada.
2. Varía el tipo de información que se recolecta de un trabajo a otro aunque utilicen la misma fuente de actividad del usuario.
3. No cuentan con verdaderos ataques realizados en las computadoras de los usuarios.
4. Algunas bases de datos no son públicas y no las pueden utilizar otros investigadores para experimentar con ellas.

Como se puede ver las bases de datos existentes tienen muchas deficiencias, principalmente en el hecho de que los ataques que tienen no son fieles a la realidad. Por ese motivo se buscó la creación de una base de datos que contenga ataques simulados pero que se acerquen mucho a ser reales, así como brindamos la posibilidad de compartir la base de datos para que otros investigadores puedan utilizarla y con ello poder comparar resultados entre clasificadores.

Tabla 1. Información básica de los usuarios de WUIL: Su labor, la versión de MS Windows que utiliza y el número de días durante los cuales se recolectó información.

Usuario	Labor	Versión de MS Windows	# de días de la bitácora
1	Administrador	Windows XP	49
2	Administrador	Windows XP	48
3	Área de Sistemas	Windows XP	35
4	Contador	Windows XP	29
5	Secretaria	Windows XP	54
6	Secretaria	Windows XP	55
7	Secretaria	Windows XP	56
8	Secretaria	Windows XP	31
9	Secretaria	Windows XP	34
10	Envíos	Windows Vista	33
11	Secretaria	Windows XP	13
12	Programador	Windows 7	44
13	Estudiante	Windows Vista	40
14	Ventas	Windows 7	34
15	Estudiante de Doctorado	Windows Vista	54
16	Estudiante de Doctorado	Windows 7	37
17	Estudiante	Windows 7	35
18	Estudiante	Windows 7	30
19	Secretaria	Windows Vista	36
20	Estudiante de Doctorado	Windows XP	23

3. La base de datos WUIL

Debido a las deficiencias que hay en las bases de datos actuales así como a que los resultados obtenidos con las fuentes de actividades que manejan no han sido concluyentes, se creó una base de datos nueva para este trabajo. La hipótesis manejada es que el análisis de la interacción del usuario con su sistema de archivos nos brinda la posibilidad de detectar intrusos. La base de datos contiene información tanto de usuarios legítimos como de tres ataques realizados en cada una de las computadoras de los usuarios legítimos.

Para validar la hipótesis se crearon bitácoras, en las cuales se guarda información de los archivos que se están accediendo. Después de procesar las bitácoras, se puede obtener información que indique qué objetos del sistema operativo está accediendo el usuario, así como la forma en que lo hace (por ejemplo, dependencias entre objetos, frecuencia de uso, etc).

3.1. Bitácoras de usuarios

Al momento de la escritura de este documento WUIL cuenta con 20 usuarios con períodos de recolección que van desde cinco hasta diez semanas, la información general de los usuarios se puede ver en la tabla 1. Este número de usuarios se espera que vaya incrementando de forma continua para enriquecer la base de datos.

Para la recolección de los datos se utilizó la herramienta Audit de Windows, la cual registra los objetos utilizados por el usuario durante su sesión. Aunque se realizó en Windows la idea general se puede realizar en cualquier Sistema Operativo. Al ser demasiadas carpetas las que hay en el Sistema Operativo completo, nos enfocamos en un par de carpetas: Escritorio y Mis Documentos.

Tabla 2. Número de accesos por cada usuario antes y después de filtrar.

Usuario	Número de accesos original	Número de accesos después de filtrar
1	293,894	272,642
2	77,608	63,536
3	40,708	15,159
4	147,418	129,847
5	350,800	338,788
6	369,440	356,194
7	352,910	335,081
8	91,852	85,482
9	178,602	178,460
10	127,532	108,643
11	547,545	545,715
12	2,692,858	197,639
13	2,307,804	2,307,253
14	1,752,639	19,812
15	454,280	453,204
16	1,444,914	1,290,295
17	245,862	213,147
18	10,409	10,315
19	90,357	54,302
20	76,943	57,608

Estas carpetas fueron seleccionadas a través de encuestas realizadas a estudiantes universitarios preguntándoles dónde buscarían información relevante de una persona si tuvieran acceso a su computadora durante 5 minutos.

Una ventaja de WUIL es que los datos fueron registrados durante el uso normal del Sistema Operativo por el usuario, y no de forma artificial como en otras bases de datos. Sin embargo, uno de los problemas al momento de guardar los registros es que guarda información que no es necesaria, como registros que no estaban contenidos en **Escritorio** o en **Mis Documentos**, por lo cual se tuvo que hacer un pre-procesamiento y se tuvo que filtrar esta información, en la tabla 2 se pueden ver los números de acceso por usuario antes y después del filtrado. Probablemente el usuario número 13 haya que dividirlo en varios usuarios debido a la cantidad de accesos que tiene aún después del filtrado, ya que en análisis previos encontramos que puede ser complicado procesar bitácoras de esas dimensiones.

3.2. Ataques y su simulación

Se simularon tres diferentes tipos de ataques que buscan abarcar ataques desde ataques que fueron hechos sin premeditación hasta ataques que fueron planeados con anticipación. Los ataques fueron simulados lo más fieles a la realidad posible, para esto se hicieron encuestas a estudiantes dónde debían indicar que pasos seguirían para realizar un ataque en una computadora la cual esta desatendida durante cinco minutos y desean obtener información sensible del usuario al que pertenece. A partir de los resultados de la encuesta se pudieron determinar los tres tipos de ataque: básico, intermedio y avanzado.

Ataque básico El ataque básico es un ataque que modela el caso de un intruso ocasional, lo que significa que trata de aprovechar una oportunidad que se le

presenta en lugar de crear él la oportunidad. En este caso se asume que el intruso no tiene consigo una memoria USB ni la posibilidad de copiar la información en ningún medio electrónico. Por lo tanto, el intruso únicamente puede abrir y cerrar archivos y si quiere extraer información únicamente puede ser utilizando su propia memoria o copiandola en papel.

Ataque intermedio En el ataque intermedio, el intruso trae una memoria USB consigo. Por lo tanto, el intruso buscará archivos que le parezcan interesantes (por ejemplo, cuenta, contraseña, o extensiones como *.doc) utilizando la herramienta de búsqueda de Windows y los copiará en su memoria USB para posteriormente revisarla y ver si hay información que le sirva.

Ataque avanzado En este caso, el intruso no sólo trae la USB si no que trae un archivo .bat que con solo ejecutarlo se encargará de realizar la búsqueda de archivos interesantes y copiarlos en la USB sin la necesidad de que el busque manualmente. Con esto el atacante se ahorra tiempo y puede robar más información.

Todos los ataques se realizaron directamente en la computadora de cada uno de los usuarios y los tres ataques dentro de una computadora fueron realizados por la misma persona. Los archivos extraídos nunca fueron revisados y se borraron después de acabar el ataque. En la tabla 3 se puede ver el número de registros por cada uno de los ataques realizados.

Finalmente, podemos decir que WUIL es una base de datos que supera algunas de las deficiencias de otras bases de datos: WUIL contiene ataques simulados muy parecidos a ataques reales y al ser una base de datos pública muchos investigadores podrán utilizarla y los resultados entre experimentos pueden ser comparables. Sin embargo, no supera algunas deficiencias, por ejemplo, la de tener tiempos de captura distintos.

4. Trabajo en proceso

Actualmente se está en el proceso de experimentar utilizando la base de datos. Para esto se considera que hay ciertos rasgos en la forma en la que interactúa el usuario con su computadora que pueden ayudar a diferenciar su comportamiento del de un atacante. A continuación explicaremos los rasgos que se busca probar que sirven para detectar intrusos.

4.1. Localidad

Hay un par de términos sobre localidad a la hora de hablar de Sistemas Operativos, en especial al hablar de las lecturas en memoria. El primer concepto es la localidad temporal, la cual argumenta que si una localidad de memoria es referenciada probablemente sea referenciada nuevamente en un período de tiempo corto. El segundo concepto es la localidad espacial y argumenta que si

Tabla 3. Bitácoras de los atacantes: número de accesos de los atacantes antes del filtrado y después del filtrado.

Usuario	Ataque 1		Ataque 2		Ataque 3	
	Accesos Totales	Accesos Filtrados	Accesos Totales	Accesos Filtrados	Accesos Totales	Accesos Filtrados
1	2,476	2,353	4,851	4,352	14,825	14,825
2	2,049	2,032	4,203	3,968	1,423	1,273
3	2,240	2,155	1,112	1,069	1,904	1,836
4	3,702	3,646	7,603	7,422	1,1369	1,1235
5	3,085	3,001	15,507	14,925	10,378	10,378
6	3,594	3,499	8,902	8,530	7,466	7,466
7	3,184	3,090	18,916	18,625	11,464	11,464
8	5,331	5,267	8,265	7,898	10,404	10,404
9	3,546	3,546	10,761	10,711	9,519	9,519
10	1,038	730	973	499	148	13
11	6033	6033	25162	25162	842	842
12	1,966	1,953	3,411	3,411	16,269	16,265
13	259	254	16098	16096	1004	1004
14	13,404	1,084	4,403	1,315	4,537	1,726
15	1,284	1,284	6,100	6,100	4,171	4,171
16	2,893	2,893	919	919	2,075	2,075
17	1,585	1,585	2,289	2,289	3,342	3,342
18	1,770	1,770	498	498	1,495	1,495
19	1,003	588	1,215	392	225	139
20	3,455	3,443	52,819	52,804	10,361	10,361

una localidad de memoria es referenciada es muy probable que otras localidades de memoria cercanas a ésta sean referenciadas en un período de tiempo corto.

Se cree que al utilizar estos conceptos, pero llevándolo a archivos y no a localidades de memoria nos puede ayudar a detectar intrusos. Se considera que en general los usuarios si se encuentran realizando un trabajo en especial, accederán archivos que se encuentran en una misma carpeta y probablemente lo hagan de forma repetida. Por otro lado se piensa que un intruso saltará entre carpetas y archivos para buscar información interesante y sin repetir archivos.

Para lograr esto es importante encontrar un valor que diga que tan cercano es un archivo de otro. Además, se debe decidir sobre una unidad de tiempo que permita ver si un archivo se está accediendo de nueva cuenta en un período de tiempo corto o no.

4.2. Tareas

Normalmente cada archivo en una computadora pertenece a una tarea de un usuario (por ejemplo, un estudiante tendrá tareas como trabajo, escuela, videojuegos, películas, pasatiempo, etc.). Se piensa que si se etiqueta cada uno de los archivos con la tarea a la que pertenecen y se analizan los cambios entre estas tareas se podran identificar patrones que ayuden a crear un perfil del usuario y con ello detectar intrusos.

El reto que es que no es posible pedirle a cada uno de los usuarios que digan cada uno de sus archivos a que tarea pertenece, por lo tanto se debe buscar algún método para etiquetar las tareas de forma automatizada. En un principio se cree que archivos contenidos dentro de una carpeta pertenecen a la misma tarea.

Después se debe crear un perfil donde se guarde la información relacionada con los cambios entre cada una de las tareas. Con esto se puede revisar nuevas transiciones entre tareas e identificar si se trata de un intruso o del usuario legítimo.

4.3. Número de accesos y número de accesos nuevos

Observando el comportamiento de los usuarios mientras interactúan con su sistema de archivos se ha notado que suelen trabajar con calma y realizando un número reducido de accesos cada cierto tiempo. Los atacantes en cambio, realizan muchos accesos en períodos de tiempo corto, revisando muchos archivos, algunos de los cuales el usuario utiliza muy poco o que en realidad nunca ha utilizado. Un acceso nuevo puede ser realizado tanto porque el archivo se acaba de crear, como porque desde que se tiene el sistema de detección activado nunca se había utilizado.

Por tal motivo, se piensa que si se revisa el número de accesos y de accesos nuevos realizados durante cierto período de tiempo, nos ayudará a detectar intrusos. Para eso hay que utilizar las bitácoras de la base de datos e ir revisando que archivos son los que utiliza el usuario y con qué frecuencia lo hace. Además hay que buscar una forma de marcar como nuevos archivos que no se han utilizado dentro de un período de tiempo largo.

5. Conclusiones y trabajo futuro

Utilizando la base de datos empleando los conceptos antes mostrados se espera obtener resultados satisfactorios en la detección de intrusiones. Además se espera que la base de datos sea utilizada por otros investigadores para poder comparar nuestros resultados con los suyos e identificar así el mejor mecanismo para detectar intrusos.

La base de datos presentada y que se está compartiendo se considera que puede ser de mucha utilidad en el área principalmente por dos motivos: primero, porque pocas veces las bases de datos son compartidas y por ende los resultados entre trabajos no son comparables; segundo, porque la base de datos contiene ataques simulados fieles a la realidad, que es algo que no se ha hecho en otros trabajos del área.

El trabajo futuro se divide en dos partes primordiales. La primera, es que además del trabajo en proceso mostrado en la sección 4 se deben buscar nuevas formas de aprovechar la base de datos. Y la segunda, se debe mantener WUIL actualizada agregando nuevos usuarios tanto de Windows como de otros sistemas operativos.

Agradecimientos Este proyecto fue apoyado parcialmente por CONACyT mediante una beca de doctorado con número de becario 241856, una beca de estancia postdoctoral a nombre de Carlos Hernández Gracidas y un proyecto de investigación básica con número 105698.

Referencias

1. Schonlau, M., DuMouchel, W., Ju, W., Karr, A., Theus, M., Vardi, Y.: Computer intrusion: Detecting masquerades. *Statistical Science* **16**(1) (2001) 58–74
2. Razo-Zapata, I., Mex-Perera, C., Monroy, R.: Masquerade attacks based on user's profile. *Journal of Systems and Software* **85**(11) (2012) 2640–2651
3. Garg, A., Rahalkar, R., Upadhyaya, S., Kwiat, K.: Profiling users in GUI based systems masquerade detection. In: *Proceedings of the 7th IEEE Information Assurance Workshop*, IEEE Computer Society Press (2006) 48–54
4. Killourhy, K.S., Maxion, R.A.: Why did my detector do *that?!* - predicting keystroke-dynamics error rates. In Jha, S., Sommer, R., Kreibich, C., eds.: *Recent Advances in Intrusion Detection*, 13th International Symposium, RAID 2010. Volume 6307 of *Lecture Notes in Computer Science.*, Springer (2010) 256–276
5. Schonlau, M.: Masquerading user data. <http://www.schonlau.net> (2008)
6. Ben-Salem, M., S., S.: Modeling user search behavior for masquerade detection. *Computer Science Technical Reports 033*, Columbia University (2010)
7. Joyce, R., Gupta, G.: Identity authentication based on keystroke latencies. *Commun. ACM* **33**(2) (February 1990) 168–176
8. Bleha, S., Slivinsky, C., Hussien, B.: Computer-access security systems using keystroke dynamics. *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on* **12**(12) (dec 1990) 1217 –1222
9. Cho, S., Han, C., Han, D.H., il Kim, H.: Web based keystroke dynamics identity verification using neural network. *Journal of Organizational Computing and Electronic Commerce* **10** (2000) 295–307
10. Haider, S., Abbas, A., Zaidi, A.: A multi-technique approach for user identification through keystroke dynamics. In: *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*. Volume 2. (2000) 1336 –1341 vol.2
11. Yu, E., Cho, S.: Ga-svm wrapper approach for feature subset selection in keystroke dynamics identity verification. In: *Neural Networks, 2003. Proceedings of the International Joint Conference on*. Volume 3. (july 2003) 2253 – 2257 vol.3
12. Araujo, L., Sucupira, L.H.R., J., Lizarraga, M., Ling, L., Yabu-Uti, J.: User authentication through typing biometrics features. *Signal Processing*, *IEEE Transactions on* **53**(2) (feb. 2005) 851 – 855
13. Kang, P., Hwang, S.s., Cho, S.: Continual retraining of keystroke dynamics based authenticator. In: *Proceedings of the 2007 international conference on Advances in Biometrics. ICB'07, Berlin, Heidelberg, Springer-Verlag* (2007) 1203–1211
14. Camiña, B., Monroy, R., Trejo, L., Sánchez, E.: Towards building a masquerade detection method based on user file system navigation. In Batyrshin, I., Sidorov, G., eds.: *Proceedings of the Mexican International Conference on Artificial Intelligence, MICAI'11*. (2011) 174–186

Procesamiento de lenguaje natural

Análisis de polaridad en Twitter

Gilberto Leon Martagón, Esaú Villatoro-Tello,
Héctor Jiménez-Salazar y Christian Sánchez-Sánchez

Departamento de Tecnologías de la Información,
División de Ciencias de la Comunicación y Diseño,
Universidad Autónoma Metropolitana Unidad Cuajimalpa, México DF
{2123805605, evillatoro, hjimenez, csanchez}@correo.cua.uam.mx

Resumen En este artículo se propone una técnica supervisada para determinar la polaridad/sentimiento de mensajes publicados en Twitter. Nuestro método propone una forma de representación vectorial, donde a partir de una selección de términos basada en la Ley de Zipf se muestra que es posible obtener resultados favorables al momento de realizar la clasificación de mensajes de acuerdo a su polaridad. La representación propuesta obedece a las características léxicas propias que surgen del uso de esta plataforma de microblogueo, así como las limitantes impuestas por la misma, tal y como el número limitado de caracteres permitidos, las etiquetas, las menciones y los enlaces Web. Los experimentos que se realizaron, sobre una colección de tuits de temáticas variadas, muestran que los atributos empleados para representar tuits permiten obtener un desempeño favorable al momento de identificar el sentimiento de los usuarios en sus publicaciones.

Palabras clave: análisis de polaridad, clasificación de textos, procesamiento de lenguaje natural, aprendizaje automático.

1. Introducción

Twitter, lanzado en 2006, sin lugar a dudas es una de las plataformas de microblogueo más importantes hoy en día. Las últimas cifras sobre su uso revelan que hay más de 200 millones de usuarios que escriben más de 400 millones de tuits cada día [1], sobre una gran variedad de temas. Debido a la aceptación que esta plataforma ha conseguido, diversas entidades, entre las que podemos contar a empresas comerciales, celebridades y políticos, se encuentran inmersas en una constante búsqueda de posicionamiento para poder influenciar de forma eficiente en la opinión de algún grupo demográfico de usuarios suscritos a Twitter.

Por lo tanto, uno de los esfuerzos involucrados en este posicionamiento consiste en obtener una idea de la opinión¹ (*i.e.*, positiva, negativa y/o neutral) sobre un tema, producto o servicio. Sin embargo, hacerlo a mano se convierte en un proceso imposible debido a la gran cantidad de información que es producida

¹ El análisis de opinión es también conocido como análisis de polaridad y/o de sentimientos.

en este tipo de redes sociales. De esto surge la necesidad de contar con herramientas automáticas que permitan hacer un análisis de polaridad de manera confiable y eficiente.

Actualmente, existen diversas técnicas para hacer análisis de polaridad en textos largos en diversos idiomas y dominios. Sin embargo, las características impuestas a través de la plataforma de Twitter, como lo es la longitud máxima de los mensajes, motivan a los usuarios a innovar en su forma de expresión, provocando con esto el surgimiento de un lenguaje propio de la plataforma, el cual varía dependiendo del idioma y del contexto (social y/o cultural) en el que estas formas de expresión son publicadas. Es importante mencionar que todo esto tienen un impacto negativo en el desempeño de técnicas tradicionales de clasificación de polaridad en textos largos, por lo tanto, cualquier intento de análisis de tuits debe tener en cuenta las características propias de Twitter.

En este trabajo proponemos una representación de las publicaciones en Twitter que toma en cuenta las características del lenguaje empleado en la plataforma. Así por ejemplo, nuestra propuesta toma en cuenta el uso de etiquetas (*hashtags*), las cuales pueden formar parte sintáctica y/o semántica de un tuit; agregado a esto también consideramos las menciones a otros usuarios, con las que ocurre lo mismo que con las etiquetas; finalmente, los enlaces a sitios Web son también considerados por nuestra forma de representación propuesta.

Adicionalmente, se propone como forma de selección de las palabras más representativas de un tuit, el uso del punto de transición, *i.e.*, se toma en cuenta la frontera entre las palabras de baja y alta frecuencia, derivado de la Ley de Zipf. Los resultados experimentales muestran que las consideraciones hechas sobre el lenguaje propio de twitter así como la forma de selección de atributos permiten obtener un desempeño favorable al momento de hacer la clasificación de polaridad.

El resto de este trabajo está organizado de la siguiente manera: la sección 2 describe brevemente el trabajo relacionado, la sección 3 describe el método propuesto mientras que la sección 4 describe los datos empleados así como los experimentos realizados. Finalmente, la sección 5 expresa las conclusiones obtenidas durante la realización de este trabajo.

2. Trabajo relacionado

En los trabajos propuestos previamente, existen algunos que toman en cuenta algunas o varias de las características propias de Twitter, como lo es el uso de emoticonos. En el trabajo propuesto en [2] de un conjunto inicial de 60 emoticonos identificados automáticamente dentro de mensajes de twitter, se decidió conservar únicamente 15 de uso común, los cuales fueron asociando manualmente con polaridades (*i.e.*, sentimientos) específicas. Una de las características principales de este subconjunto de emoticonos es que son representados a partir de pocos caracteres ASCII y cuentan con un significado claro y preciso, *e.g.*, los mostrados en las primeras dos columnas de la Tabla 1.

El problema de considerar un conjunto de emoticonos con estas características es que no refleja la variedad de emoticonos que se usan actualmente en la plataforma de twitter. Es normal encontrar emoticonos de más de 5 caracteres que representan una idea más compleja y menos específica que aquellas que reflejan los emoticonos simples. De esta forma, cualquier intento de considerar emoticonos para hacer análisis de polaridad debe contar con un diccionario extenso de estos nuevos emoticonos, los cuales a su vez pueden combinarse para generar nuevas expresiones y/o emociones (Tabla 1), lo cual no es el propósito del presente trabajo.

Tabla 1. Comparación de emoticonos simples VS emoticonos compuestos.

Emoticonos Simples	Significado	Emoticonos Compuestos	Significado
:)	Alegría		Persona enojada lanzado un escritorio
:(Tristeza		Persona cantando
:)	Guiño		Persona cantando
:D	Risa		Amigos chocando manos
XD	Burla		Oso inexpresivo

Además de la limitante anterior, hay muchos tuits que utilizan uno o más emoticonos de manera irónica, resultando en una publicación que contiene una idea contraria a la polaridad definida por el emoticono. Dado esto, sistemas como el propuesto en [2] requieren considerar información del contexto para poder hacer una adecuada clasificación.

Algunos otros trabajos consideran las palabras contenidas en los tuits para realizar la clasificación de polaridad [3,4,5]. En éstos se concluye que la forma más eficiente de representación de los tuits es por medio del uso de unigramas de palabras en vez de bigramas. Sin embargo, una debilidad de este tipo de trabajos es que realizan experimentos sobre dominios y/o temáticas bien definidas, y además dependen de algún recurso léxico.

La propuesta presentada en este artículo, a diferencia de trabajos previos, no considera el uso de emoticonos, ni ningún otro tipo de recurso léxico para la clasificación de polaridad en tuits. Nuestra propuesta considera que el uso de las etiquetas (*#hashtags*), las menciones a otros usuarios (*@Usuario*) y los enlaces a páginas Web que aparecen dentro de los tuits pueden ser útiles para determinar la polaridad de los mismos. Agregado a esto, mostramos que considerando las palabras cercanas al punto de transición, es posible lograr un desempeño adecuado en la clasificación de la polaridad de tuits de temáticas diversas.

3. Método propuesto

3.1. Preprocesamiento de los tuits

Previo al proceso de entrenamiento y clasificación se realizó un preprocesamiento a los tuits el cual consistió en los siguientes pasos:

- Los tuits se convierten a minúsculas con la finalidad de normalizar el vocabulario.
- Cualquier secuencia de espacios en blanco se convierte en un solo espacio.
- Cualquier mención a usuarios encontrada se convierte a la cadena “PARA-USUARIO”, de modo que se conserve la intención de las menciones, pero como un único elemento del vocabulario.
- Cualquier enlace encontrado se convierte a la cadena “ENLACE-SALIENTE”, de modo que se conserve la intención de los enlaces, pero como un mismo elemento del vocabulario. Es importante mencionar que no se hace uso de la información contenida en cada sitio, sin embargo, se cree que el uso de esta información puede enriquecer el modelo propuesto.
- Se eliminan los signos de puntuación. Esto también elimina cualquier emoticono que pudiera aparecer, puesto que no fueron tomados en cuenta para el funcionamiento de este modelo.
- Cada palabra de un tuit es llevada a su raíz léxica. Este proceso se lleva a cabo mediante el lematizador Porter [6].
- Se eliminan las palabras vacías y/o funcionales.

3.2. Forma de representación de los tuits

En este trabajo se aborda la problemática del análisis de polaridad en tuits desde el paradigma de clasificación de textos (CT)². Bajo este paradigma un primer paso necesario es el *indexado* de los documentos de entrenamiento (Tr), actividad que denota hacer el mapeo de un documento d_j en una forma compacta de su contenido. La representación más comúnmente utilizada para representar cada documento es un vector con términos ponderados como entradas, concepto tomado del modelo de espacio vectorial usado en recuperación de información [8]. Es decir, un texto d_j es representado como el vector $\vec{d}_j = \langle w_{kj}, \dots, w_{|\tau|j} \rangle$, donde τ es el *diccionario*, *i.e.*, el conjunto de términos que ocurren al menos una vez en algún documento de Tr , mientras que w_{kj} representa la importancia del término t_k dentro del contenido del documento d_j . En ocasiones τ es el resultado de filtrar las palabras del vocabulario, *i.e.*, resultado de un preprocesamiento (Sección 3.1). Una vez que hemos hecho los filtrados necesarios, el diccionario τ puede definirse de acuerdo a diferentes criterios, sin embargo el que se empleó en esta propuesta corresponde a la Bolsa de Palabras.

² La Clasificación de Textos es la tarea de asociar automáticamente categorías predefinidas con documentos a partir del análisis de su contenido [7].

La Bolsa de Palabras (BOW)³ es la forma tradicionalmente utilizada para representar los documentos [7]. Este método de representación utiliza a las palabras simples como los elementos del vector de términos.

Con respecto al peso w_{kj} , se tienen diferentes formas de calcularlo, entre las más usadas en la comunidad científica se tienen el ponderado booleano y ponderado por frecuencia relativa de términos. Una breve descripción es dada a continuación:

- *Ponderado Booleano*: Consiste en asignar el peso de 1 si la palabra ocurre en el documento y 0 en otro caso.

$$w_{kj} = \begin{cases} 1, & \text{si } t_k \in d_j \\ 0, & \text{en otro caso} \end{cases} \quad (1)$$

- *Ponderado por frecuencia relativa (TF-IDF)*: Este tipo de ponderado es una variación del tipo anterior y se calcula de la siguiente forma:

$$w_{kj} = TF(t_k) \times IDF(t_k) \quad (2)$$

donde $TF(t_k)$ es la frecuencia del termino t_k en el documento d_j . IDF es conocido como la “frecuencia inversa” del termino t_k dentro del documento d_j . El valor de IDF es una manera de medir la “rareza” del termino t_k . Para calcular el valor de IDF se utiliza la siguiente ecuación:

$$IDF(t_k) = \log \frac{|D|}{|\{d_j \in D : t_k \in d_j\}|} \quad (3)$$

donde D representa la colección de documentos que está siendo indexada.

Este tipo de técnicas trae un costo agregado, que es el producir un espacio de términos (atributos) τ de alta dimensionalidad (i.e., $|\tau| \rightarrow \infty$). Este problema puede ocasionar problemas de *sobre-ajuste* en el proceso de aprendizaje, i.e., ocurre el fenómeno por medio del cual un clasificador se adapta a las características contingentes de Tr , en lugar de únicamente a las características constitutivas de las categorías, provocando problemas de efectividad debido a que el clasificador tiende a comportarse mejor sobre los datos con los que ha sido entrenado y sin conservar la tendencia en aquellos no vistos.

Uno de los métodos que ha mostrado ser efectivo como técnica de reducción de dimensionalidad es por medio de conservar los términos que se encuentran alrededor del punto de transición (pt_T) [9]. El tp_T es un valor de frecuencia que divide a los términos del vocabulario τ en dos conjuntos de términos, los de baja y alta frecuencia [10,11]. En los estudios realizados en [9,10,11] se demuestra que los términos de frecuencia media están fuertemente relacionados con el contenido de los documentos, lo cual permite resolver efectivamente tareas como la clasificación temática de textos. Nuestra hipótesis es que por medio de utilizar

³ En inglés se conoce como Bag Of Words.

un subconjunto de términos de frecuencias medias a altas es posible clasificar la polaridad en los tuits. La forma tradicional de calcular el punto de transición es:

$$tp_T = \frac{\sqrt{8 * I_1 + 1} - 1}{2} \quad (4)$$

donde I_1 representa el número de palabras con frecuencia 1 en el vocabulario τ .

3.3. Métodos de clasificación

El clasificador Naïve Bayes (*NB*) se considera como parte de los clasificadores probabilísticos, los cuales se basan en la suposición que las cantidades de interés se rigen por distribuciones de probabilidad, y que la decisión óptima puede tomarse por medio de razonar acerca de esas probabilidades junto con los datos observados [12]. Para nuestro trabajo empleamos el Naïve Bayes tradicional, el cual se describe a continuación.

En este esquema el clasificador es construido usando Tr para estimar la probabilidad de cada clase. Entonces, cuando una nueva instancia (documento) d_j es presentada, el clasificador le asigna la categoría $c \in C$ más probable por aplicar la regla:

$$c = \arg \max_{c_i \in C} P(c_i | d_j) \quad (5)$$

utilizando el teorema de Bayes para estimar la probabilidad tenemos:

$$c = \arg \max_{c_i \in C} \frac{P(d_j | c_i) P(c_i)}{P(d_j)} \quad (6)$$

dado que el denominador en la ecuación anterior no difiere entre categorías puede omitirse quedando de la siguiente forma:

$$c = \arg \max_{c_i \in C} P(d_j | c_i) P(c_i) \quad (7)$$

tomando en cuenta que el esquema es “naïve” (*i.e.*, se presupone la independencia entre atributos), se asume que las características son condicionalmente independientes dadas las clases. Esto simplifica los cálculos produciendo:

$$c = \arg \max_{c_i \in C} P(c_i) \prod_{k=1}^n P(d_{kj} | c_i) \quad (8)$$

donde $P(c_i)$ es la fracción de documentos en Tr que pertenecen a la clase c_i , es decir:

$$P(c_i) = \frac{|Tr_{c_i}|}{|Tr|} \quad (9)$$

y $P(d_{kj} | c_i)$ se calcula de acuerdo a:

$$P(d_{kj} | c_i) = \frac{1 + |Tr_{ki}|}{|A| + \sum_{l=1}^{|A|} |Tr_{li}|} \quad (10)$$

donde $|Tr_{ki}|$ es el número de documentos dentro del conjunto de entrenamiento que poseen el atributo k y además pertenecen a la clase c_i (i.e., d_{ki}), $|A|$ es el número total de atributos, de esta forma $|Tr_{li}|$ es el número de ejemplos de entrenamiento que poseen el atributo l y pertenecen a la clase c_i .

En resumen, la tarea de aprendizaje en el clasificador Naïve Bayes consiste en construir una hipótesis por medio de estimar las diferentes probabilidades $P(c_i)$ y $P(d_{kj}|c_i)$ en términos de sus frecuencias sobre Tr .

3.4. Evaluación

Para evaluar un sistema de clasificación de texto se utilizan las medidas de *Precisión* y *Recuerdo*, que son medidas comunes en el área de recuperación de información. La precisión (P) es la proporción de documentos clasificados correctamente en una clase c_i con respecto a la cantidad de documentos clasificados en esa misma clase. El recuerdo (R), la proporción de documentos clasificados correctamente en una clase c_i con respecto a la cantidad de documentos que realmente pertenecen a esa clase. Así, la precisión se puede ver como una medida de la corrección del sistema, mientras que el recuerdo da una medida de cobertura o completitud.

Adicionalmente, es común emplear la medida- F para describir el comportamiento de la clasificación, la cual se define como:

$$medida - F = \frac{(1 + \beta^2)Precision * Recuerdo}{\beta^2 Precision + Recuerdo} \quad (11)$$

donde con $\beta = 1$ representa la media armónica entre la precisión y el recuerdo. La función de β es la de controlar la importancia relativa entre las medidas de precisión y recuerdo. Es común asignar un valor de 1 indicando igual importancia a ambas medidas.

4. Pruebas realizadas

En esta sección se describen el conjunto de datos empleados para la realización de nuestros experimentos así como la configuración empleada en cada caso propuesto. Es importante mencionar que para todos los experimentos se empleó como método de clasificación el algoritmo de Naïve Bayes.

4.1. Conjunto de datos

Para la realización de nuestros experimentos trabajamos con los Tuits proporcionados por el SemEval 2013⁴. Los tuits proporcionados por el foro de evaluación representan mensajes de una gran variedad de temas entre los cuales se encuentra información de personajes públicos, productos y eventos.

⁴ <http://www.cs.york.ac.uk/semeval-2013/task2/>

Los tuits proporcionados para entrenamiento están etiquetados en tres clases, *positivo*, *negativo* y *neutral*. A diferencia de trabajos previos, el trabajar con esta colección representa un reto mayor pues el problema de clasificación no es binario. La distribución de los datos de entrenamiento fue de 950 tuits por clase. Estos datos fueron utilizados para entrenar y generar nuestros modelos de clasificación.

Agregado a los datos de entrenamiento, el SemEval 2013 proporcionó un conjunto de datos de prueba (2,845 en total) distribuidos de la siguiente manera: 740 negativos, 975 positivos, y 1,135 neutros. Estos datos fueron utilizados para evaluar nuestra propuesta.

4.2. Método base

Como método base se empleó una forma de representación la típica BOW empleando un esquema de pesado *booleano* (Sección 3). El objetivo de este experimento fue determinar la complejidad del problema de clasificación de polaridad por medio de técnicas tradicionales de CT.

4.3. Experimentos

Para los siguientes experimentos se utilizó como forma de representación una BOW empleando un esquema de pesado TF-IDF. Al mismo tiempo se utilizó como estrategia de reducción de dimensionalidad la selección de los términos alrededor del tp_T . En la tabla 2 aparecen como renglones: el porcentaje de frecuencia tomado alrededor del tp_T , la frecuencia más baja considerada, la frecuencia más alta considerada, y en el último renglón el tipo de frecuencias usadas por cada experimento.

Tabla 2. Resultados obtenidos bajo las diferentes configuraciones propuestas.

Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
23 %	23 %	79 %	89 %	94 %
69	69	20	10	5
117	759	759	759	759
medias	media-alta	media-baja	baja	muy-bajas

Es importante mencionar que sólo para el experimento 1 se utilizó el rango alrededor del tp_T para definir los límites inferiores y superiores de la frecuencia de los términos, mientras que para los experimentos 2 a 5 el límite superior se dejó abierto a la frecuencia más alta, *i.e.*, 759.

4.4. Resultados

La tabla 3 muestra los resultados obtenidos de los experimentos planteados. La primer columna hace referencia al nombre del experimento, mientras que la

Tabla 3. Resultados obtenidos bajo las diferentes configuraciones propuestas.

Nombre experimento	Num. atributos	Tipo pesado	Precisión (P)	Recuerdo (R)	Medida F_β
Base	7,620	Booleano	0.35	0.35	0.35
Exp 1	22	TF-IDF	0.36	0.35	0.35
Exp 2	41	TF-IDF	0.44	0.44	0.44
Exp 3	238	TF-IDF	0.54	0.54	0.54
Exp 4	539	TF-IDF	0.59	0.60	0.60
Exp 5	1,399	TF-IDF	0.64	0.65	0.65

segunda y tercer columna indican el esquema de pesado empleado en la representación y el número de atributos (términos) considerados en la representación.

Como es posible observar en la Tabla 3 las técnicas tradicionales de CT (*i.e.*, Experimento 1) no resultan ser apropiadas para el problema de clasificación de polaridad. Este hecho es mostrado una vez más con los resultados obtenidos en el Experimento 2, donde al usar el punto de transición en su forma tradicional, se obtiene un desempeño similar al Experimento 1, mostrando con esto que para identificar la polaridad de los tuits es necesario incluir términos que van más allá del contenido de los mensajes.

Los resultados del Experimento 2 muestran que al ampliar el rango de los términos alrededor del punto de transición es posible mejorar los resultados de clasificación. Estos resultados nos motivaron a hacer experimentos ampliando el rango de frecuencias bajas. Como es posible ver en los resultados del Experimento 5, por medio de utilizar tan sólo un vector de atributos de 1399 elementos (18.35% del tamaño original del vector de términos) es posible alcanzar un valor de *medida* – $F = 0,65$, lo que significó una mejora de más de un 82% respecto de la configuración base.

5. Conclusiones

La clasificación de polaridad de tuits comparte algunos elementos y características que tiene la clasificación de polaridad de textos más largos. Sin embargo, hay suficientes diferencias como para que las técnicas útiles en el segundo caso necesiten ser ajustadas para probar ser eficientes en el contexto de Twitter.

Como puede verse en los experimentos realizados, la representación adecuada de cada elemento a clasificar es crucial para esta labor, y esto involucra tanto el tamaño de la representación como la calidad de la información que contenga.

El método base, que utiliza una BOW *binaria* resulta en un modelo con mucha información poco relevante, que tarda mucho en entrenarse y que produce un modelo de clasificación altamente sesgado. Por el contrario, el método propuesto aprovecha la cantidad de información que cada palabra puede aportar en el caso de cada tuit en relación al conjunto de tuits en general, por lo que se muestra como una mejor solución para el problema de la clasificación de polaridad automática.

Nuestros experimentos mostraron que es necesario definir formas automáticas para definir el número de atributos que se utilizarán para la representación de

los elementos a clasificar, ya que, por un lado, un número pequeño genera un conjunto de atributos poco informativos para el modelo de clasificación. Por otro lado, si el número de atributos es demasiado grande, estos atributos introducen ruido que resulta en un sesgo significativo del clasificador, resultando en un desempeño deficiente.

Los resultados obtenidos nos motivan a seguir trabajando sobre esta línea. Como trabajo futuro se planea experimentar con atributos más complejos, por ejemplo bi-gramas de palabras, con los cuales se cree que la representación de los tuits puede ser enriquecida. Agregado a esto, se considera experimentar con el uso de los emoticonos, los cuales han mostrado aportar información en trabajos previos.

Agradecimientos. Agradecemos a la Universidad Autónoma Metropolitana Unidad Cuaajimalpa y al proyecto SEP-PROMEP número 48510294 (UAM-CA-31) por el apoyo para la asistencia a este evento.

Referencias

1. Wickre, K. (2013). Celebrating #Twitter7. En <http://blog.twitter.com/2013/03/celebrating-twitter7.html> (Última visita en Marzo 21 de 2013.)
2. Davidiv, D., Tsur, O. y Rappoport, A. (2010). Enhanced Sentiment Learning Using Twitter Hashtags and Smileys. En *Proceedings of the 23rd International Conference on Computational Linguistics: Posters (Coling 2010)*. pp. 241-249
3. Pang, B., Lee, L., y Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. En *Proceedings of the Empirical methods in natural language processing EMNLP'02*. pp. 79-86.
4. Pang, B. y Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. En *Proceedings de la Association for Computational Linguistics*. pp. 271-278.
5. Sidorov, G. , Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., Díaz-Rangel, I., Suárez-Guerra, S., Treviño, A., y Gordon, J. (2012) Empirical Study of Machine Learning Based Approach for Opinion Mining in Tweets. LNAI Vol. 7629, pp. 1-14.
6. Porter , M. F. (1997) An algorithm for suffix stripping. Morgan Kaufmann Publishers Inc. pp. 313-316.
7. Sebastiani F. (2002) Machine Learning in Automated Text Categorization. En *ACM Computing Surveys*, Vol. 34, No. 1, March 2002, pp. 1-47.
8. Baeza-Yates, R., y Ribeiro-Neto, B. (1999) Modern Information Retrieval, Addison Wesley.
9. Reyes-Aguirre, B., Moyotl-Hernández, E., y Jiménez-Salazar, H. (2003) Reducción de términos índice usando el punto de transición. En *Avances en Ciencias de la Computación*. pp. 127-130.
10. Zipf, G. K.(1949) Human Behavior and the Principle of Least-Effort, Addison-Wesley, Cambridge MA.
11. Booth, A. D. (1967) A law of occurrences for words of low frequency. En *Information and Control*. Vol. 10(4). pp. 386-393.
12. Mitchell, T. (1997) Machine Learning. McGraw-Hill.

Aplicación de la implicación textual en un sistema de la validación de respuestas automáticas

Partha Pakray¹, Alexander Gelbukh² y Sivaji Bandyopadhyay¹

¹ Jadavpur University, India

² Instituto Politécnico Nacional, México

www.parthapakray.com, www.gelbukh.com,
www.sivajibandyopadhyay.com

Resumen. La tarea de la respuesta automática a preguntas consiste en obtener de forma automática las respuestas a las preguntas del usuario basándose en una gran colección de documentos o la Internet abierta. Se diferencia de la tarea de recuperación de información en que su resultado es una o varias respuestas cortas y no un conjunto de documentos. Uno de los pasos en el proceso de respuesta automática a preguntas es la validación de la respuesta: la evaluación de la idoneidad de una posible respuesta para la pregunta específica. Dados dos textos llamados T (texto) y H (hipótesis), la tarea de reconocimiento de implicación textual consiste en decidir si el significado de H se puede deducir lógicamente del significado de T. Hemos aplicado dos técnicas diferentes de implicación textual a la tarea de la validación de la respuesta en el contexto de la tarea de respuesta automática a preguntas. En nuestros experimentos con el conjunto de prueba en inglés AVE 2008, se obtuvo la puntuación $c@1$ de 0.65 y 0.69, respectivamente.

Palabras clave: respuesta automática a preguntas, implicación textual, validación de la respuesta, reconocimiento de entidades nombradas.

An Application of Textual Entailment Recognition in an Answer Validation System

Abstract. The question answering task consists in automatically obtaining answers to the user's questions basing on a large collection of documents or the open Internet. It differs from the information retrieval task in that the result is one or several short answers and not a set of documents. One of the steps in the question answering process is answer validation: evaluation of the appropriateness of a candidate answer for the question. Given two texts called T (Text) and H (Hypothesis), the textual entailment recognition task consists in deciding whether the meaning of H can be logically inferred from that of T. We applied the two different textual entailment techniques to the answer validation task in the context of question answering. In our experiments on the AVE 2008 English test set, we obtained the $c@1$ score of 0.65 and 0.69, respectively.

Keywords: question answering, textual entailment, answer validation, named entity recognition.

1. Introducción

El reconocimiento de la implicación textual (RIT) [1] es uno de los desafíos recientes del Procesamiento del Lenguaje Natural (PLN) que tiene un amplio potencial de las aplicaciones en otras tareas del PNL tales como la generación automática de resúmenes, paráfrasis, extracción de información, traducción automática y respuesta automática a preguntas.

La tarea del RIT se define como una relación direccional entre pares de expresiones textuales, denotados T –el “texto”, lo implicante–, y H –la “hipótesis”, lo implicado. Si el significado de H se puede inferir a partir del significado de T entonces T implica H. La tarea de RIT consiste entonces en decidir automáticamente si tal relación existe o no entre los T y H dados.

Se han organizado seis competiciones internacionales sobre esta tarea: RTE-1 a RTE-7 (por las siglas en inglés, “*Recognizing Textual Entailment*”) en los años 2005 a 2011, respectivamente. El trabajo aquí presentado, en sus versiones anteriores, ha participado en las RTE-5 y RTE-6.

La tarea de la validación de la respuesta está dirigida a la creación de los sistemas que miden la exactitud de la respuesta a una pregunta, dado un texto de soporte. Específicamente, tales sistemas reciben un conjunto de tripletas: pregunta, respuesta y el texto de soporte, y para cada tal tripleta devuelven una de las tres posibles respuestas: “seleccionado”, “validado” o “rechazado”. Aquí las respuestas rechazadas

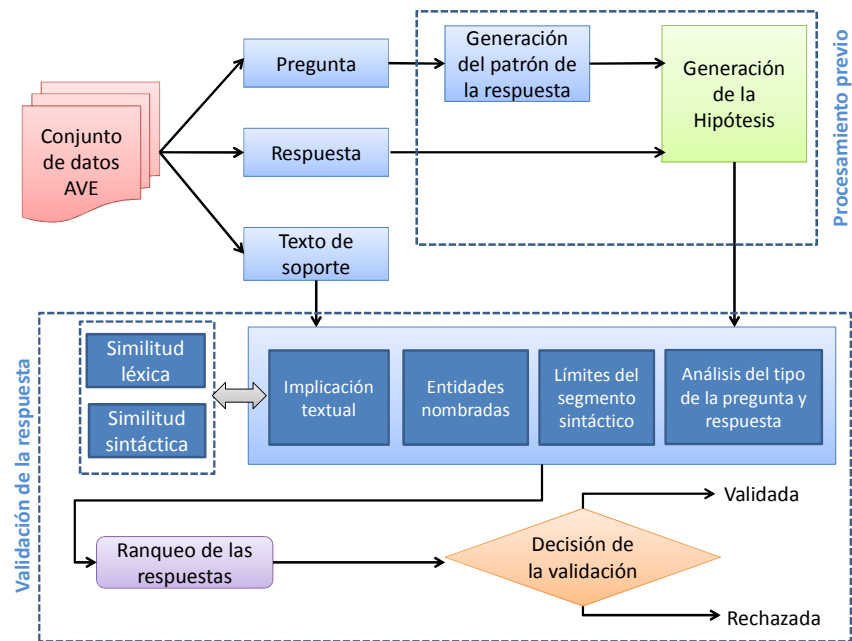


Figura 1. Arquitectura del sistema de la validación de respuestas


```

<q id="0061" leng="EN">
  <q_str>¿Dónde nació Joseph Fourier?</q_str>
  <a id="0061_1" valor="rechazado">
    <a_str>Paris</a_str>
    <t_str doc="Joseph Fourier">Joseph Fourier Joseph Fourier Joseph Fourier Jean
    Baptiste Joseph Fourier Nacido 21 de marzo 1768 Auxerre, Yonne, France Fallecido el 16
    de mayo 1830 París, Francia Residencia Francia Nacionalidad francesa Campo
    Matemático, físico e historiador Institución École Normale École Polytechnique Alma
    Mater École Normale Asesor Doctoral Joseph Lagrange Estudiantes de doctorado Gustav
    Dirichlet Giovanni Plana Conocido por transformada de Fourier Religión Católica Jean
    Baptiste Joseph Fourier (21 marzo 1768 — 16 mayo 1830) fue un matemático y físico
    francés quien es mejor conocido por iniciar la investigación de la serie de Fourier y sus
    aplicaciones a los problemas de flujo de calor.</t_str>
  </a>
  <a id="0061_2" valor="desconocido">
    <a_str>Francia</a_str>
    <t_str doc="Joseph Fourier">Vida Fourier nació en Auxerre, en el
    departamento de Yonne de Francia, hijo de un sastre.</t_str>
  </a>
  <a id="0061_3" valor="rechazado">
    <a_str>Grenoble</a_str>
    <t_str doc=" Universidad Joseph Fourier">Universidad Joseph Fourier Universidad
    Joseph Fourier Université Joseph Fourier (Universidad Joseph Fourier) es una universidad
    francesa, situada en la ciudad de Grenoble, centrada en los campos de las ciencias, las
    tecnologías y la salud.</t_str>
  </a>
  <a id="0061_8" valor="rechazado">
    <a_str>Auxerre</a_str>
    <t_str doc="Auxerre.html">Auxerre.</t_str>
  </a>
  ....
</q>

```

Figura 1. Traducción de un ejemplo de un documento del conjunto de documentos del ejercicio de validación de respuesta (AVE 2008).

son las que no son apropiadas para la pregunta, validadas son apropiadas, y una de las validadas se elige como seleccionada, y será ésta la que será devuelta al usuario. Se han organizado tres competiciones internacionales en esta tarea: AVE 2006 [2] (por sus siglas en inglés, “*Answer Validation Exercise*”), AVE 2007 [3] y AVE 2008 [4].

Hemos desarrollado un sistema de implicación textual en el escenario monolingüe y el de idiomas diferentes. Estamos aplicando nuestra técnica del RIT en los sistemas de respuesta a preguntas y de validación de la respuesta. Para la tarea de RIT hemos desarrollado tanto sistemas basados en reglas como sistemas basados en el aprendizaje automático que tienen en cuenta los rasgos semánticos, sintácticos y léxicos. Nos dirigimos principalmente a establecer la relación de implicación textual (IT) basándonos en las medidas de similitud léxica, sintáctica y semántica.

Tabla 1. Detección del límite del segmento sintáctico (traducción literal del inglés).

Ejemplo 1	Pregunta	¿Qué fue la nacionalidad de Jacques Offenbach?
	Analizador de categorías gramaticales de Stanford	¿/. Qué/WP fue/VBD la/DT nacionalidad/NN de/IN Jacques/NNP Offenbach/NNP ?/.
	Analizador de segmentos sintácticos con los campos aleatorios condicionales	Qué/WP/B-NP fue/VBD/B-VP la/DT/B-NP nacionalidad/NN/I-NP de/IN/B-PP Jacques/NNP/B-NP Offenbach/NNP/I-NP
	Límite del segmento	(Qué/WP/B-NP) (fue/VBD/B-VP) (la/DT/B-NP#nacionalidad/NN/I-NP) (de/IN/B-PP) (Jacques/NNP/B-NP#Offenbach/NNP/I-NP#)
Ejemplo 2	Pregunta	¿Dónde fue Joseph Fourier nacido?
	Analizador de categorías gramaticales de Stanford	¿/. Dónde/WRB fue/VBD Joseph/NNP Fourier/NNP nacido/VBN ?/.
	Analizador de segmentos sintácticos con los campos aleatorios condicionales	Dónde/WRB/B-ADVP fue/VBD/B-VP Joseph/NNP/B-NP Fourier/NNP/I-NP nacido/VBN/B-VP
	Límite del segmento	(Dónde/WRB/B-ADVP) (fue/VBD/B-VP) (Joseph/NNP/B-NP#Fourier/NNP/I-NP) (nacido/VBN/B-VP#)

Para eso, hemos usado la detección de las entidades nombradas, categorías gramaticales, similitud basada en el diccionario WordNet, similitud basada en segmentos sintácticos (*chunks* en inglés), similitud basada en n-gramas, similitud basada en la distancia léxica, similitud sintáctica, similitud semántica y otras para decidir si existe la relación de IT con aproximaciones basadas en reglas y en el aprendizaje de máquina, tratando el problema de RIT como un problema de clasificación.

Con estos desarrollos, hemos participado en diferentes competencias internacionales de evaluación que se concentraron principalmente en las cuestiones de la clasificación en dos vías y la clasificación en vías múltiples, en ambos casos tanto en el escenario monolingüe como en el escenario de idiomas distintos (*cross-lingual* en inglés).

En la tarea de RIT de la competencia principal de la Conferencia de Análisis de Texto (TAC, por sus siglas en inglés) en su quinta edición (2009) propusimos un sistema basado en léxico [5]. En la tarea RIT de la competencia principal de la TAC en su sexta edición (2010) propusimos un sistema basado en la similitud léxica y sintáctica [6]. Hemos desarrollado otro sistema [7] que se basa en la evaluación de similitud entre pares T-H en lugar de las etiquetas de implicación.

Hemos utilizado el Lenguaje de Redes Universal (UNL, por sus siglas en inglés: Universal Networking Language, www.undl.org) para identificar los rasgos semánticos. UNL tiene todos los componentes de un lenguaje natural. Hemos desarrollado un sistema de RIT basado en UNL que compara las relaciones de UNL en el texto y en la hipótesis.

Tabla 2. Regla de la generación del patrón para el tipo de pregunta “¿Qué” (traducción literal del inglés).

Pregunta tipo “¿Qué”		
Regla 1	¿Qué VP NP?	¿Qué es astronomía?
Patrón	NP VP </Respuesta>	Astronomía es </Respuesta>
Regla 2	¿Qué NP VP NP?	¿Qué vehículo es usado en el deporte “Moto cross”?
Patrón	</Respuesta> VP NP	</Respuesta> es usado en el deporte “Moto cross”
Regla 3	¿Qué <ser-verbo> <el nombre de>/<la nacionalidad de>/<la ocupación de> NP?	¿Qué fue la nacionalidad de Jacques Offenbach?
Patrón	<el nombre de> NP <ser-verbo> </Respuesta>	La nacionalidad de Jacques Offenbach fue </Respuesta>
Regla 4	¿<PP> qué NP <ser-verbo> NP VP?	¿En qué países es la navegación deportiva sobre hielo practicado?
Patrón	NP <ser-verbo> VP PP </Respuesta>	La navegación deportiva sobre hielo es practicada en </Respuesta>
Regla 5	¿PP qué NP1 <ser-verbo> NP2 VP?	¿En qué año fue Emerson Lake & Palmer formada?
Patrón	NP2 VP PP </Respuesta>	Emerson Lake & Palmer se formó en </Respuesta>
Regla 6	¿PP qué NP1 <ser-verbo> que NP2 VP NP3?	¿En qué año fue que James Hurt ganó el mundial de Formula Uno?
Patrón	NP2 VP NP3 PP </Respuesta>	James Hurt ganó el mundial de Formula Uno en </Respuesta>
Regla 7	¿PP que NP1 <ser-verbo> NP2 VP?	¿En qué año será el Hotel Brunwick City en Glasgow abierto?
Patrón	NP2 <ser-verbo> VP PP </Respuesta>	El Hotel Brunwick City en Glasgow será abierto en </Respuesta>

En este artículo presentamos nuestro siguiente paso: un sistema de validación de respuesta [8] basado en la RIT. Primero, combinamos la pregunta y la respuesta en la hipótesis H, y consideramos el texto de soporte como el texto T, para comprobar la relación de IT como validada o rechazada. Hemos aplicado la técnica de IT al sistema de validación de las respuestas para tomar decisiones de validación o rechazo. También hemos aplicado la técnica de IT en las tareas de respuesta automática a preguntas (por ejemplo, QA4MRE) en CLEF 2011 [9] y CLEF 2012 [10].

2. La arquitectura de nuestro sistema de validación de respuestas

Hemos construido un sistema de validación de respuesta mostrado en la figura 1, donde la validación se basa en la IT. Las características importantes que se utilizaron para el desarrollo del sistema son la generación del patrón, la generación de la

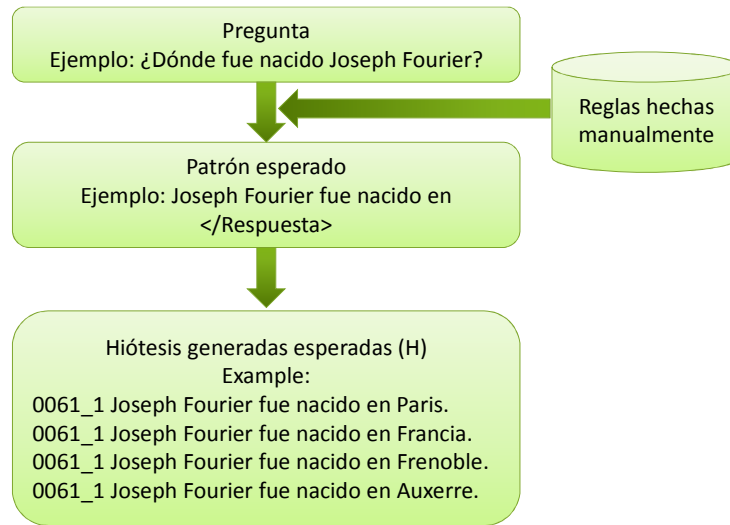


Figura 2. Un ejemplo de la generación de la hipótesis.

hipótesis, RIT, el reconocimiento de entidades nombradas, análisis de tipo de la pregunta-respuesta, el módulo de límites del segmento sintáctico (*chunk*) y el módulo de la similitud sintáctica. El sistema está basado en reglas. En primer lugar, combinamos la pregunta y la respuesta en la Hipótesis (H), y el texto de soporte lo consideramos como el Texto (T), con lo cual obtenemos las decisiones de la IT como validación o rechazado.

2.1. Conjunto de datos

El experimento se ha llevado a cabo principalmente con el conjunto de datos AVE 2008. Un ejemplo de este conjunto de datos se muestra en la figura 2.

El sistema analiza la pregunta usando el etiquetador de las categorías gramaticales de Stanford (<http://nlp.stanford.edu/software/tagger.shtml>) y el analizador de segmentos sintácticos (*chunker*) basado en los campos aleatorios condicionales (CRF, por sus siglas en inglés, <http://crfchunker.sourceforge.net>). Después de este procesamiento, el sistema detecta los límites de los segmentos sintácticos, como se muestra en la tabla 1.

En este módulo, cada pregunta es convertida en un patrón de respuesta afirmativa en la ranura de la respuesta (</Respuesta>). Las reglas para el patrón de respuesta correspondiente a cada pregunta fueron desarrollados hecha manualmente. La generación de un patrón para la pregunta de tipo “¿qué” (“what” en inglés) se muestra en la tabla 2.

Tabla 3. Respuesta sustituida por la regla de generación de patrones.

N del par	Hipótesis generadas esperadas
0061_1	Joseph Fourier fue nacido en Paris.
0061_2	Joseph Fourier fue nacido en Francia.
0061_3	Joseph Fourier fue nacido en Grenoble.
0061_8	Joseph Fourier fue nacido en Auxerre.

2.3. Módulo de la generación de la hipótesis

Después de la generación del patrón, la plantilla `</Respuesta>` se sustituye por la cadena de la respuesta (`<a_str>`) que forma la hipótesis generada. Ahora, el sistema tiene el Texto (T), el texto de apoyo y la Hipótesis (H), la hipótesis generada. Por ejemplo, para la pregunta número 0061 en el conjunto de prueba AVE-2008, el sistema genera las hipótesis para cada una de las respuestas alternativas como se describe en la tabla 3.

Un ejemplo de la generación de la hipótesis se muestra en la figura 3.

2.4. Módulo del reconocimiento de la implicación textual

El módulo más importante de nuestro sistema es el módulo de la IT. El proceso de RIT se basa en la implicación léxica y sintáctica, es decir, es un proceso de RIT denominado híbrido. En esta sección describimos un sistema de RIT de dos vías (validación o rechazo) que utiliza las características léxicas y sintácticas. La arquitectura del sistema se muestra en la figura 4. Nuestro sistema híbrido de RIT utiliza el aprendizaje basado en la máquina de soporte vectorial, con treinta y cuatro rasgos usados para el entrenamiento: cinco rasgos del proceso del RIT léxico, diecisiete rasgos de la distancia léxica y once rasgos del sistema del RIT de dos vías basado en la información sintáctica. Los cinco rasgos léxicos se describen en detalle en [4] y los once rasgos sintácticos se describen en detalle en [11].

Los rasgos de similitud léxica que se utilizan en el sistema actual para comparar la hipótesis con el texto son:

- comparación de unigramas basada en WordNet,
- comparación de bigramas,
- subsecuencia común más larga,
- salta-gramas (skip-grams),
- comparación de las bases morfológicas.

Los rasgos de la distancia léxica, véanse los detalles en [12], son los siguientes:

- distancia de bloque,
- derivación de la longitud de Chapman,
- longitud media de Chapman,
- similitud del coseno,
- similitud de Dice,
- distancia euclidiana,
- similitud de Jaccard,
- distancia de Jaro,
- distancia de Jaro-Winkler,
- coeficiente de coincidencia,
- distancia de Monge-Elkan,
- distancia de Needleman-Wunch,

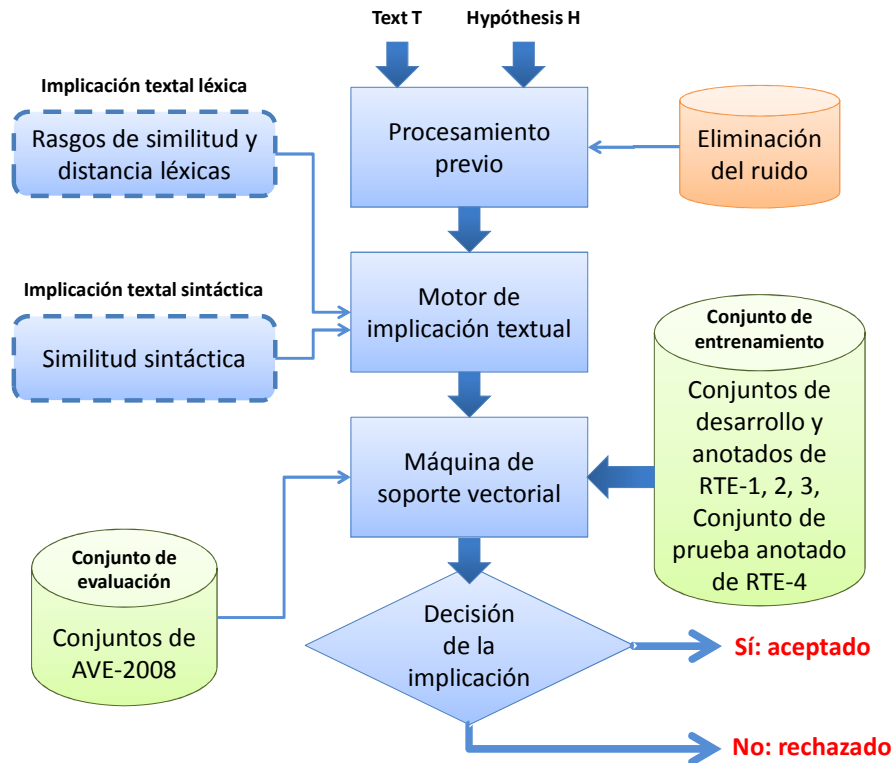


Figura 3. La arquitectura del sistema de reconocimiento de la implicación textual.

- coeficiente de traslape,
- distancia en qgramas,
- distancia de Smith-Waterman,
- distancia de Smith-Waterman con ir-a,
- distancia de Smith-Waterman con ir-a de ventanas afina.

Para el RTE basado en la información sintáctica, se usaron los siguientes rasgos:

- comparación del sujeto con el sujeto,
- comparación del sujeto y el verbo basada en WordNet,
- comparación del sujeto y el objeto,
- comparación del objeto y el verbo,
- comparación del objeto y el verbo basada en WordNet,
- comparación del sujeto y el objeto cruzada,
- comparación del número,
- comparación de los sustantivos,
- comparación de las frases preposicionales,
- comparación de los determinantes,
- comparación de otras relaciones.

Tabla 4 Entidades nombradas reconocidas por la herramienta LT-TTT2 (traducción literal del inglés).

Para la entidad nombrada	Etiqueta dada por la herramienta LT-TTT2	Ejemplo
UBICACIÓN	<enamex tipo="ubicación">	India, Francia etc.
ORGANIZACIÓN	<enamex tipo="organización">	BBVA Bancomer etc.
PERSONA	<enamex tipo="persona">	Henry Purcell etc.
FECHA	<timex tipo="fecha">	24 de junio de 1982 etc
TIEMPO	<timex tipo="tiempo">	La tarde etc
PORCENTAJE	<numex tipo="porcentaje">	5% etc
DINERO	<numex tipo="dinero">	\$7 millones etc
PAPEL	<papel>	Capitán, Presidente etc.
NÚMERO	<phr c="cd">	Dos, 6 etc

El sistema utiliza la biblioteca LIBSVM (<http://www.csie.ntu.edu.tw/~cjlin/libsvm>) para construir el archivo del modelo. El sistema del RIT utiliza los siguientes conjuntos de datos: el conjunto de desarrollo y de prueba de la RTE-1, los conjuntos de desarrollo y los conjunto de prueba anotados de las RTE-2, RTE-3 y el conjunto de prueba anotado de la RTE-4 para la tarea de la clasificación de dos vías para el entrenamiento para construir el archivo del modelo. La LIBSVM utiliza un clasificador para aprender de este conjunto de datos. Para el entrenamiento fueron utilizados 3967 pares de texto y hipótesis. Después del entrenamiento del sistema lo evaluamos en el conjunto de datos de AVE. Como el resultado, el sistema produce las decisiones del RIT: “sí” –validar– o “no” – rechazar–.

2.5. Módulo del reconocimiento de las entidades nombradas

Este módulo se basa en la detección y comparación de las entidades nombradas en el par del texto de soporte (T) y la hipótesis (H) generada. Una vez se han detectados las entidades nombradas de la hipótesis y el texto, el siguiente paso es determinar el número de las entidades nombradas en la hipótesis que coinciden con las del texto correspondiente. La medida de coincidencia NE_Match se define como sigue:

$$NE_Match = \frac{\text{número de entidades nombradas en común en T y H}}{\text{número de entidades nombradas en la H}}$$

Si el valor de NE_Match es 1, entonces todas las entidades nombradas en la hipótesis coinciden con las en el texto. En este caso el par de texto y hipótesis se considera como un caso de implicación y al par T y H se le asigna el valor “validado”. De lo contrario, a este par se le asigna el valor “rechazado” con una puntuación de coincidencia.

Para el reconocimiento de las entidades nombradas, usamos el llamado componente de entidad nombrada LT-TTT2 (<http://www.ltg.ed.ac.uk/software/lt-ttt2>). Este componente de software reconoce y marca las siguientes clases de las entidades nombradas: expresiones numéricas (por ejemplo, cantidades de dinero y porcentajes), expresiones temporales (por ejemplo, fechas y horas), expresiones de entidad

Tabla 5. Tipo de respuesta esperada por el tipo de pregunta.

Tipo de pregunta	Respuesta esperada
¿Quién	PERSONA
¿Cuándo	FECHA / TIEMPO
¿Dónde	UBICACIÓN
¿Qué	OBJETO
¿Cómo	MEDIDA

nombrada (por ejemplo, personas, organizaciones y lugares) y papeles (por ejemplo, *presidente*).

Un ejemplo de la entidad designada por la herramienta el LT-TTT2 se muestra en la tabla 4.

2.6. Módulo del análisis del tipo de pregunta y respuesta

Las preguntas iniciales son preprocesadas utilizando el analizador sintáctico de dependencias de Stanford (<http://nlp.stanford.edu/software/lex-parser.shtml>). El tipo de la pregunta y el tipo de la respuesta esperada se identifican generalmente por la palabra principal de la pregunta, por ejemplo “¿qué”, “¿quien”, “¿dónde”, “¿cuándo”, etc. La tabla 5 muestra preguntas y el tipo de la respuesta correspondiente que se espera.

Por ejemplo, si el tipo de pregunta es “¿dónde”, el tipo de la respuesta que se espera es una *ubicación*. La cadena de respuesta “<a_str>” es analizada sintácticamente por el componente LT-TTT2 antes mencionado. Si el componente LT-TTT2 genera la etiqueta “<entidad_nombrada tipo=ubicación>”, entonces la cadena de respuesta que es validada, de lo contrario es rechazada. Un ejemplo se muestra en la tabla 6.

Para el tipo “¿qué” buscamos la palabra clave (por ejemplo, una *empresa*) relacionada con la palabra “¿qué” a través de una relación de dependencia. Si la palabra clave es una *empresa*, entonces el tipo de la respuesta esperada es “organización”. Si la cadena de la respuesta correspondiente está marcada por la herramienta LT-TTT2 como “<entidad_nombrada tipo=organización>”, entonces la cadena de la respuesta se marca como validada, de lo contrario, como rechazada.

Si el tipo de pregunta es “¿cómo” y la cadena de respuesta se marcada como “<frase c=cd>” por la herramienta LT-TTT2, entonces la cadena de respuesta se marca como validada, de lo contrario, como rechazada.

Para cada comparación, se genera una puntuación cuantitativa. Un ejemplo de la validación a través del reconocimiento de las entidades nombradas se muestra en la tabla 7. Por último, el sistema otorga una puntuación final de coincidencia.

2.7. Módulo de la detección de los segmentos sintácticos (chunks)

Las oraciones con las preguntas son preprocesados utilizando el analizador sintáctico de dependencias de Stanford. Las palabras junto con su la información de la

Tabla 6. Análisis con la herramienta LT-TTT2.

Pregunta	¿Dónde la Olimpiada de Invierno tendrá lugar en 1994?
Respuesta esperada	UBICACIÓN
Hipótesis	La Olimpiada de Invierno de 1994 tendrá lugar en Lillehammer.
Hipótesis analizada	<timex tipo="fecha">Invierno</timex> <timex tipo="fecha">1994</timex> <enametipo="ubicación">Lillehammer</enametipo>

categoría gramatical se pasan al analizador de los segmentos sintácticos (*chunker*) basado en un campo aleatorio condicional (*English CRF Chunker*) para extraer de las preguntas los segmentos sintácticos de nivel de frase, por ejemplo, frase nominal (NP), frase verbal (VP), etc. Hemos desarrollado un módulo basado en reglas para identificar los límites de las frases. Las frases clave son identificadas para cada pregunta. Las frases que se relacionan con cada preposición constituyen la frase clave que corresponde a la relación de preposición. Cada frase verbal presente en la pregunta es también una frase clave. Si no hay relaciones preposicionales en la pregunta, entonces todas las frases son consideradas como frase clave, excepto la frase que forma la pregunta: “¿qué”, “¿dónde”, etc. Estas frases clave se buscan en el texto de soporte asociado con la oración de la pregunta. Un ejemplo de una pregunta con la información sobre los límites del segmento sintáctico (frase), las relaciones de dependencia y un conjunto de frases clave extraídos se muestran en la tabla 8.

El texto de soporte se analiza con el analizador sintáctico de dependencias de Stanford. La salida del analizador se pasa al analizador de los segmentos sintácticos basado en un campo aleatorio condicional (CRF) para extraer los segmentos de nivel de frase para cada oración en el texto de soporte. Las frases clave identificadas en la pregunta están luego comparados con las del texto de soporte asociado a la pregunta. Si un segmento clave (frase) completo coincide, heurísticamente el peso asignado a la coincidencia se define como sigue:

$$W_f = \frac{\text{longitud del segmento} + K}{\text{longitud del texto}},$$

donde K es el número de los segmentos (frases), para dar mayor peso a la coincidencia completa de los segmentos. Si existe una coincidencia parcial de tales segmentos, el peso se define sigue:

$$W_p = \frac{\text{longitud parcial del segmento}}{\text{longitud del texto}}.$$

El peso de un par de pregunta y el texto de soporte se identifica como el total de los pesos de los segmentos (frases) clave. Los pares de pregunta y el texto de soporte que reciben el peso máximo se identifican y las respuestas correspondientes se etiquetan como validados. If el par de la pregunta y el texto de soporte recibe un peso cero, entonces es etiquetado como rechazado, con una puntuación de coincidencia.

Tabla 7. Validación del reconocimiento de las entidades nombradas.

Pregunta	¿Dónde fue construido el Volkswagen Polo Playa?		
Respuesta	UBICACIÓN		
N del par	Respuesta	Entidad nombrada	Calificación de la respuesta
0108 1	Sudáfrica	Ubicación	validada
0108 2	Sucesor		rechazada
0108 3	Apuntador		rechazada
0108 9	Tom Ennis	Persona	rechazada
0108 11	Ibiza		rechazada
0108 12	Playa		rechazada
0108 13	Eurona	Ubicación	validada

2.8. Módulo de la calificación cuantitativa de las respuestas

En este módulo, el sistema ya tiene los pesos generados por el módulo de RIT, el módulo del reconocimiento de las entidades nombradas, el módulo de análisis del tipo de la pregunta y el módulo de la detección de los límites de los segmentos sintácticos.

2.8.1. Módulo de la decisión sobre la validación de la respuesta

En ese módulo, el sistema finalmente decide cuál de las respuestas es correcta. Para la identificación de la respuesta correcta el sistema utiliza las siguientes dos técnicas.

Técnica 1 En esta variante, el sistema se usa una técnica de votación. Primero, el sistema compruebe si alguna entidad nombrada está presente en la hipótesis H generada. Se verifican las siguientes condiciones:

1. Si en la hipótesis H generada está presente una entidad nombrada, entonces el sistema comprueba el resultado del módulo del RIT de las entidades nombradas. Si este módulo indica “validado”, entonces se comprueban los resultados del módulo de la implicación textual, el módulo del análisis del tipo de la pregunta y respuesta, y el módulo de la segmentación sintáctica. Si todos estos módulos indican “validado”, entonces la respuesta dada a la pregunta inicial se marca como validada. De lo contrario, la respuesta se etiqueta como rechazada.
2. Si en la hipótesis H generada, no existen entidades nombradas, entonces se comprueban los resultados del módulo del RIT, módulo de análisis del tipo de la pregunta y respuesta, módulo de análisis de los segmentos sintácticos y el módulo de la similitud sintáctica. Si todos estos módulos generan el resultado “validado”, entonces esta respuesta a la pregunta inicial se marca como validada, de lo contrario, la respuesta es etiquetada como rechazada.

Tabla 8. Extracción de los segmentos sintácticos (*chunks*, traducción literal del inglés).

Pregunta	¿Qué fue la nacionalidad de Jacques Offenbach?
Límite del segmento (frase)	(Qué/WP/B-NP) (fue/VBD/B-VP) (la/DT/B-NP nacionalidad/NN/I-NP) (de/IN/B-PP) (Jacques/NNP/B-NP Offenbach/NNP/I-NP)
Dependencias	[attr fue Qué det nacionalidad la nsubj fue nacionalidad nn Offenbach Jacques prep_of nacionalidad Offenbach]
Frase clave extraída	(fue/VBD/B-VP) (la/DT/B-NP nacionalidad/NN/I-NP) (Jacques/NNP/B-NP Offenbach/NNP/I-NP)

Técnica 2 Para cada texto e hipótesis, el sistema calcula la puntuación de los módulos del RIT, de análisis de tipo de pregunta y respuesta, de la implicación a través de las entidades nombradas y de la similitud de los segmentos sintácticos. A continuación, el sistema considera la puntuación más alta para validar, y los pares restantes son etiquetados como rechazados.

3. Resultados experimentales

Hemos experimentado con nuestro sistema de validación de respuestas usando el conjunto de desarrollo para inglés del AVE 2008. Este conjunto consiste de 195 pares, de los cuales sólo 21 son positivos (10.77% del número total de los pares). Los valores del recuerdo (*recall*), precisión y la medida F obtenidos sobre este corpus para las respuestas marcadas como validadas se muestran en la tabla 9.

El conjunto anotado de prueba para el inglés del AVE 2008 consiste de 1055 pares, y el número de las respuestas marcadas como validadas es 79 (7.5% del total). Los valores del recuerdo (*recall*), precisión y la medida F obtenidos sobre este corpus para las respuestas marcadas como validadas se muestran en la tabla 10.

4. Conclusiones y trabajo futuro

La respuesta automática a preguntas es una nueva forma de la recuperación de información, mucho más cómoda para el usuario y mucho más apropiada para la comunicación humano-computadora [13]. La validación de la respuesta es un paso importante en el proceso de la respuesta automática a preguntas. Con este trabajo hemos demostrado que las técnicas del reconocimiento de la implicación textual (RIT)

Tabla 9. Conjunto de datos de desarrollo del AVE

Conjunto de datos de desarrollo del AVE	Resultados con Técnica 1	Resultados con Técnica 2
“validado” en el conjunto de desarrollo	21	21
“validado” por nuestro sistema	34	32
coincidencia en la marca “validado”	15	18
Precisión	0.44	0.56
<i>Recall</i>	0.71	0.85
Medida F	0.54	0.68

Tabla 10. Conjunto de datos de prueba del AVE.

Conjunto de datos de prueba del AVE	Resultados con Técnica 1	Resultados con Técnica 2
“validado” en el conjunto de prueba	79	79
“validado” por nuestro sistema	78	75
coincidencia en la marca “validado”	52	54
Precisión	0.66	0.72
<i>Recall</i>	0.65	0.68
Medida F	0.65	0.69

se pueden usar con éxito para la validación y ponderación de las posibles respuestas en el contexto de la respuesta automática a preguntas.

Hemos presentado en detalle la arquitectura del sistema de validación de la respuesta, la cual consiste en varios módulos desde el procesamiento previo de los datos hasta el reconocimiento de la implicación textual y la toma final de la decisión. Los resultados experimentales confirman nuestra hipótesis en cuanto a la utilidad de los métodos de RIT para la validación de la respuesta.

En nuestro trabajo futuro, planeamos experimentar con otros rasgos para el aprendizaje automático, así como con representaciones del texto más ricas, tales como los grafos conceptuales, n-gramas sintácticos [14] etc. Finalmente, intentaremos aplicar a nuestro trabajo las recientes técnicas de extracción abierta de los hechos [15] los cuales, como pensamos, pueden jugar el papel de los segmentos sintácticos pero mucho mejor. Esta generalización es aún más importante dado que existen sistemas para la extracción de los hechos no sólo para el inglés, sino también para el español [16].

Agradecimientos. Este trabajo fue parcialmente patrocinado por el Gobierno de México y el Gobierno de la India a través del proyecto DST India – CONACYT México 122030 / Mexico-India 2011–2014 “*Answer Validation through Textual Entailment*”, la Unión Europea a través del 7o Programa Marco, proyecto 269180 “*Web Information Quality Evaluation Initiative*” (WIQ-EI) y el Gobierno de México a través del Instituto Politécnico Nacional, proyecto SIP 20131702.

Referencias

1. Dagan, I., Glickman, O., Magnini, B. The PASCAL Recognising Textual Entailment Challenge. En: Proc. PASCAL Challenges Workshop on RTE, 2005.
2. Peñas, A., Rodrigo, Á., Sama, V., Verdejo, F. Overview of the answer validation exercise 2006. En: Working Notes, CLEF 2006.
3. Peñas, A., Rodrigo, Á., Verdejo, F. Overview of the answer validation exercise 2007. En: Working Notes, CLEF 2007.
4. Rodrigo, Á., Peñas, A., Verdejo, F. Overview of the answer validation exercise 2008. En: Working Notes, CLEF 2008.
5. Pakray, P., Bandyopadhyay, S., Gelbukh, A. Lexical based two-way RTE System at RTE-5. En: System Report, TAC RTE Notebook, 2009.
6. Pakray, P., Pal, S., Poria, S., Bandyopadhyay, S., Gelbukh, A. JU_CSE_TAC: Textual Entailment Recognition System at TAC RTE-6. En: System Report, Text Analysis Conference Recognizing Textual Entailment Track (TAC RTE) Notebook, 2010.
7. Pakray, P., Barman, U., Bandyopadhyay, U., Gelbukh, A. A Statistics-Based Semantic Textual Entailment System. En: Lecture Notes in Artificial Intelligence, 7094, Springer, 2011, p. 267–276.
8. Pakray, P., Barman, U., Bandyopadhyay, S., Gelbukh, A. Semantic Answer Validation using Universal Networking Language. En: International Journal of Computer Science and Information Technologies, 3(4):4927–4932, 2012.
9. Pakray, P., Bhaskar, P., Banerjee, S., Chandra Pal, B., Bandyopadhyay, S., Gelbukh, A., A Hybrid Question Answering System based on Information Retrieval and Answer Validation. En: CLEF 2011 Workshop on Question Answering For Machine Reading Evaluation (QA4MRE). CLEF 2011 Labs and Workshop. Notebook Papers. Amsterdam. 2011, 16 pp.
10. Bhaskar, P., Pakray, P., Banerjee, S., Banerjee, S., Bandyopadhyay, S., Gelbukh, A. Question Answering System for QA4MRE@CLEF 2012. En: CLEF 2012 Workshop on Question Answering For Machine Reading Evaluation (QA4MRE). CLEF 2012 Labs and Workshop. Notebook Papers, Roma, Italia, 2012.
11. Pakray, P., Gelbukh, A., Bandyopadhyay, S., A Syntactic Textual Entailment System Using Dependency Parser. En: Lecture Notes in Computer Science, N 6008, 2010, Springer, pp. 269–278.
12. Neogi, S., Pakray, P., Bandyopadhyay, S., Gelbukh, A. JU_CSE_NLP: Language Independent Cross-lingual Textual Entailment System. En: *SEM: First Joint Conference on Lexical and Computational Semantics. Collocated with NAACL-HLT, 2012, June 7–8, Montreal, Canadá, 2012.
13. Sidorov, G. Desarrollo de una aplicación para el diálogo en lenguaje natural con un robot móvil. En: Research in computing science, 56, 2012.
14. Sidorov, G. N-gramas sintácticos y su uso en la lingüística computacional, Vectores de investigación, 2013.

Partha Pakray, Alexander Gelbukh y Sivaji Bandyopadhyay

15. Aguilar-Galicia, H.; Sidorov, G.; Ledeneva, Y. Extracción automática de hechos de libros de texto basada en estructuras sintácticas. En: *Research in computing science*, 55:15–26, 2012.
16. Zhila A., Gelbukh A. Comparison of open information extraction for English and Spanish. En: *Computational Linguistics and Intelligent Technologies*, 12(19):714–722, 2013.

Desarrollo de una aplicación para el diálogo en lenguaje natural con un robot móvil

Grigori Sidorov

Laboratorio de lenguaje natural y procesamiento de texto,
Centro de Investigación en Computación (CIC),
Instituto Politécnico Nacional (IPN), México DF,
México

www.cic.ipn.mx/~sidorov

Resumen. En este artículo presentamos tareas de desarrollo y avances de un proyecto que desarrolla todas las fases de la comunicación con un robot móvil en el mundo real. Se integra el software de reconocimiento y síntesis de voz (Dragon Naturally Speaking SDK), el software de manejo de un robot móvil en la arquitectura cliente-servidor, el software de análisis lingüístico (FreeLing), y el software que implementa un modelo de estrategias de diálogo. Las estrategias de diálogo consisten en varios patrones que se obtienen después de la aplicación del análisis lingüístico de las frases de entrada basándose en la detección de las palabras específicas. El software es aplicable al robot móvil tipo *Pioneer* que es un estándar de facto en el campo de la robótica móvil (de los robots que no son de juguete), aunque sin mayores cambios el software puede ser aplicado a otros tipos de robots.

Palabras clave. Robótica móvil, diálogo en lenguaje natural, estrategias de diálogo.

1. Introducción

La robótica móvil es una línea de investigación que tiene un gran potencial en su desarrollo y aplicaciones. En su etapa moderna, cuando ya existen los robots de una buena calidad que permiten su navegación autónoma, es importante poder transmitir los comandos a esos robots a través de la voz, igual que recibir la respuesta del robot a través del mismo medio—la voz, y en general establecer la comunicación natural con el robot..

El uso de la voz permite entablar un diálogo con el robot que imita un diálogo con otro ser humano, y de esa manera la comunicación se vuelve más natural. Para que

este diálogo fuera coherente, aunque se realizase sobre un tema específico—en nuestro caso, sobre los movimientos de mismo robot—, se necesita implementar las siguientes etapas de procesamiento por parte del robot:

- Reconocer la frase que se le manda (la voz),
- Procesar la respuesta en un esquema tradicional de un procesador lingüístico [1],
- Preparar la respuesta según las estrategias de diálogo específicas,
- Generar la frase de respuesta según las reglas gramaticales,
- Pronunciar la respuesta, y
- Realizar la acción solicitada.

En este artículo presentamos un proyecto que desarrolla todas las fases de esta comunicación, integrando:

- Módulo de reconocimiento y síntesis de voz (*Dragon Naturally Speaking SDK*),
- Módulo de manejo de un robot móvil en la arquitectura cliente-servidor,
- Módulo de análisis lingüístico (basado en Freeling¹), y
- Módulo que implemente un modelo de estrategias de diálogo.

Las estrategias de diálogo consisten en varios patrones que se obtienen después de la aplicación del análisis lingüístico de las frases de entrada basándose en la detección de las palabras específicas y sus combinaciones. Eso claramente es una simplificación de un proceso real de diálogo, pero su uso es justificado para los fines prácticos, es decir, para tener un robot reaccionando correctamente a los comandos. Con esos patrones se determina a qué actos de habla corresponden las oraciones y se actúa en consecuencia.

El software es aplicable al robot móvil tipo *Pioneer* que es un estándar de facto en el campo de la robótica móvil, en este caso sin contar los robots del tamaño real, y no los que son de juguete. Sin embargo, sin mayores cambios el software desarrollado puede ser aplicado a otros tipos de robots. En este caso se cambiaría el módulo de la interfaz con el robot de nivel bajo, nada más, es decir, la manera de como transmitir al robot los comandos de movimientos, etc.

El robot móvil tipo *Pioneer* es un robot que se mueve sobre las ruedas, sin embargo, las mismas ideas son aplicables a los robots humanoides (antropomorfos), que utilizan piernas para su movimiento—ya que el hecho de usar piernas en lugar de ruedas claramente no afecta las estrategias de desplazamiento como tal a nivel de planeación y ejecución global.

¹ Freeling es un analizador que resuelve con muy buena calidad muchos problemas de procesamiento automático de lenguaje natural (también para el español), como análisis morfológico, sintáctico, resolución de anáfora, detección de entidades nombradas [19].

2. Descripción del robot

Un robot móvil es un dispositivo que está provisto de los sensores de medición de distancia hasta los obstáculos, tiene las ruedas y la manera muy precisa de medir la distancia recorrida, puede ejercer los comandos de una computadora usando la interfaz serial RS-232, dar vueltas (a un ángulo específico) y moverse hacia atrás o adelante con una velocidad determinada, y obviamente pararse. En sí, no parece que son muchas opciones pero son las características suficientes para modelar el comportamiento relacionado con el movimiento humano en un espacio restringido. En este caso no se usan los módulos de visión, siendo sustituidos por la medición de distancias hasta los obstáculos.

El “cerebro” del robot es una computadora estándar, que realiza todo el procesamiento, y manda al robot los comandos correspondientes. Como es algo que debe moverse junto con el robot y no depender de los cables eléctricos, usualmente esta computadora es un laptop.

Cabe mencionar que el robot tiene dos tipos de sensores: ocho sonares y un sensor laser. El sensor laser obtiene un gran número de lecturas y realmente permite la detección de obstáculos muy precisa, haciendo el uso de los sonares redundante para las aplicaciones prácticas.

Como es usual en el campo de robótica, un robot móvil puede moverse de manera adecuada si tiene de antemano un mapa de su ambiente, entonces una de las tareas previas a su uso es desarrollar este mapa y presentarlo al usuario. Es lógico porque un robot móvil solamente puede medir las distancias hacia los obstáculos; la visión para la robótica móvil es una rama que se está desarrollando rápidamente, pero todavía no tiene los avances necesarios para realizar los movimientos confiables solamente basándose en visión.

Un problema del robot es la imposibilidad de detectar a los obstáculos desnivelados, especialmente los precipicios o escaleras hacia abajo, dado que sus sensores solo “barren” en un plano. Por lo mismo, en el mapa se puede marcar ciertas áreas desniveladas como las áreas prohibidas, y el robot no va a poder pasar por allá y no caería en un precipicio.

En nuestro caso usamos a un robot móvil tipo *Pioneer 3-DX*, aunque el desarrollo es aplicable a todos los robots similares.

3. Los actos de habla

Un diálogo es un intercambio de réplicas de por lo menos dos sujetos. Cada sujeto tiene su modelo del mundo y da un propósito a cada réplica suya, expresando alguna de sus intenciones.

Existen muchos trabajos sobre diálogos en los sistemas automáticos de respuestas, por ejemplo, para consultas de horarios y precios de los boletos, como un ejemplo, mencionamos a Torres (2006) [4]. Sin embargo, nuestro interés principal es un diálogo más complejo, que involucre a un robot.

Roy y Reiter (2005) [9] mencionan algunos problemas fundamentales en la interacción/diálogo con los robots:

- uso de sistemas de representación multimodal,
- asociación de palabras con las acciones y con las percepciones,
- modelado del contexto,
- detección de la granularidad correcta de los modelos,
- integración de modelos y planeación temporales,
- habilidad de mapear las experiencias pasadas, es decir aprendidas, con la interacción actual,
- habilidad de entender y tomar en cuenta la perspectiva de humanos.

En general, podemos afirmar que existen dos tipos de contextos (uno de ellos con dos dimensiones) del diálogo [11]:

- contexto lingüístico,
- contexto extralingüístico:
 - dimensión temporal,
 - dimensión de conocimiento del mundo.

Del otro lado, la estrategia de diálogo puede formalizarse [6, 7] como un proceso de decisión secuencial en términos de mapeo entre posibles estados y posibles acciones, es decir, como una serie de transiciones entre estados a través de acciones. En este caso, la idea sería minimizar el costo de las transiciones entre esos dos conjuntos, donde los nodos son estados del robot y las acciones son las transiciones: para llegar a un estado específico se puede usar varias rutas. Sin embargo, es un problema exponencial. Como siempre en las tareas de optimización exponenciales, se puede buscar la optimización de esas rutas con algún algoritmo de inteligencia artificial, como, por ejemplo, un algoritmo evolutivo.

Una de las posibles soluciones a las estrategias del diálogo mencionadas en [4], también es actual para los robots. Por estrategia de diálogo se entiende el conjunto de decisiones tomadas por el módulo gestor de diálogo que permiten:

- Evitar que el usuario se sienta perdido, explicando las acciones del sistema.
- Responder directamente a las preguntas del usuario.
- Permitir al usuario que corrija al sistema (posibilidad de negociación), por las causas de tanto errores del sistema como por el cambio de decisión del mismo usuario.

En este caso existe la negociación, también llamada la iniciativa mixta, cuando tanto el robot como el humano pueden iniciar la interacción. Cabe mencionar que este trabajo se orienta en sistemas tipo telefónicas, cuando no existe un mundo en común para los sujetos de interacción.

Analizando el lado relacionado con los actos del habla presentado por el creador de esta teoría J. Searle [12]:

- Afirmaciones (*assertives*), actos de habla que afirman la verdad, por ejemplo, profesando.
- Directivas (*directives*), que deben causar alguna acción del oyente: peticiones, comandos, consejos.
- Promesas (*commissives*), que el hablante contrae los compromisos futuros.
- Expresiones (*expressives*) que expresan la emoción o la posición, como felicitaciones, excusas, agradecimientos.
- Declaraciones (*declarations*), que cambian algo en el mundo haciendo una declaración, es decir, que son los performativos.

Se puede observar, que son los actos de habla demasiado generales para su aplicación directa en un modelo práctico de diálogo. Por lo que normalmente se proponen unas clasificaciones más detalladas y específicas.

Otras estrategias generales interesantes más prácticas que pueden ser aplicadas en un sistema específico [14] son:

- Considera las fuentes disponibles de información lo más temprano posible,
- Mantén las hipótesis alternativas y posterga la decisión con una sola opción lo más largo posible.

En la comunicación existen otros principios muy relevantes con respecto a las estrategias conocidas como Máximas de Grice [13], que básicamente son las presuposiciones en una interacción cooperativa exitosa:

1. Máxima de calidad: sé honesto.
 - No digas algo que crees que es falso.
 - No digas algo de que no tienes evidencias.
2. Máxima de cantidad: cantidad de información debe ser adecuada.
 - Haz tú contribución tan informativa como es necesario (para los propósitos dados de intercambio de información).
 - No seas más informativo de lo que es necesario.
3. Máxima de relación (relevancia): sé relevante (es claro intuitivamente para una persona pero es difícil de determinar cómo cumplirlo).
4. Máxima de forma de expresión (manera): sé claro.
 - Evita la obscuridad de expresión.

- Evita la ambigüedad.
- Sé breve.
- Sé ordenado.

Las máximas son aplicables al diálogo, pero son más bien unas reglas generales que modelos prácticos como, por ejemplo, [15, 17]. En un diálogo práctico existen las obligaciones y conocimiento común (*common ground*). Por ejemplo, si el usuario pide algo al sistema, el sistema obtiene la obligación de cumplir eso durante el desarrollo de la interacción, o bien informar que no se puede. Las obligaciones se definen como relaciones entre los diferentes actos de habla.

El conocimiento común garantiza que los participantes comparten el mismo conjunto de creencias e intenciones en cada momento. Esto garantiza que se entenderán correctamente, es decir, en correspondencia a la intención de otra parte. En la práctica a menudo se pierde el conocimiento común y debe restablecerse.

Es necesario recordar manejar correctamente de cambio de turnos, y el cambio o continuación del tópico de conversación. Cabe mencionar que antes de continuar con el otro tema, debe existir la evidencia que de alguna manera terminamos el tema anterior, es conocido como principio de clausura (*principle of closure*).

De manera muy general cualquier diálogo completo sigue al siguiente esquema:

Saludo—Apertura—Negociación/Acción—Finalización—Despedida.

Los actos de habla a menudo pueden tener unos actos de habla de respuesta que les corresponden. Por ejemplo, Levinson [16] menciona las siguientes pares de actos y las reacciones típicas:

- petición → aceptar / rechazar
- oferta → aceptar / rechazar
- invitación → aceptar / rechazar
- evaluación → acuerdo / desacuerdo
- pregunta → respuesta esperada / respuesta inesperada
- acusación → negación / admisión
- disculpa → minimización
- agradecimiento → minimización (ej., “de nada”)
- saludo → saludo

Existen varios modelos de actos de habla en diálogos. Por ejemplo, un modelo muy conocido es DAMSL de Allen y Gore [17], que contiene 4 dimensiones, cada uno incluye algunas subdimensiones y los actos de habla.

1. Estado comunicativo (no operable, abandonado, o monólogo vs. diálogo).
2. Nivel de información.
Actos de habla:

- tarea,
 - mantenimiento de tarea,
 - mantenimiento de comunicación.
3. Análisis hacia adelante (*Forward Looking Function*): afecta las acciones subsecuentes.
- Actos de habla:
- Declaración (*statement*),
 - Compromiso de la acción subsecuente (*committing-speaker-future-action*),
 - Convencional (*conventional*),
 - Performativo (*explicitperformative*),
 - Exclamación (*exclamation*),
 - Otro (lo que dice el hablante): suposición, confirmación,
 - Influencia de la acción subsecuente (*influencing-addressee-future-action*) (como en la teoría de Searle):
 - Una sugerencia o una lista de opciones,
 - Un comando real,
 - Petición de información.
4. Análisis hacia atrás (*Backward Looking Function*): relación con el discurso anterior.
- Actos de habla:
- Concordancia (*agreement*): aceptar, rechazar, mantener comunicación,
 - Entendimiento (*understanding*),
 - Respuesta (*answer*) que corresponde a la petición de información.

Si se trata de los actos de habla de manera más específica, se puede mencionar varios conjuntos de los siguientes actos de habla que se presentan, por ejemplo, en [18]:

- Agradecer (*Thank*)
- Saludar (*Greet*)
- Introducir (*Introduce*)
- Despedirse (*Bye*)
- Solicitar_comentario (*Request_comment*)
- Sugerir (*Suggest*)
- Rechazar (*Reject*)
- Aceptar (*Accept*)
- Solicitar_sugerencia (*Request_suggest*)
- Iniciar (*Init*)
- Argumentar (*Give_reason*)
- Retroalimentar (*Feed_back*)

- Deliberar (*Deliberate*)
- Confirmar (*Confirm*)
- Clarificar (*Clarify*)
- Desviarse (*Digress*)
- Motivar (*Motivate*)
- Desechar (*Garbage*)

Si vamos a considerar alguna clasificación de los actos de habla, una manera de clasificarlos es:

- Toma de turno (*Turn-taking*)
 - Tomar turno (*take turn*)
 - Mantener turno (*keep turn*)
 - Liberar turno (*release turn*)
 - Asignar turno (*assign turn*)
- Mantenimiento (*Grounding*)
 - Confirmar (*acknowledge*)
 - Reparar (*repair*)
 - Continuar (*continue*)
- Básico (*Core*)
 - Informar (*inform*)
 - Preguntar algo específico (*wh-question*)
 - Aceptar (*accept*)
 - Requerir (*request*)
 - Ofrecer (*offer*)
- Argumentación (*Argumentation*)
 - Elaborar (*elaborate*)
 - Resumir (*summarize*)
 - Pregunta-respuesta (*question-answering*)
 - Clarificar (*clarify*)

4. Tareas de desarrollo

Como ya mencionamos anteriormente, el objetivo general de nuestro desarrollo se puede formular de la siguiente manera: desarrollar un software que permita dialogar con un robot móvil sobre sus movimientos utilizando el lenguaje natural. La implementación del software incluye la integración de todos los módulos, lo que según nuestra experiencia no es una tarea trivial. Nótese que es muy difícil para un robot de “entender” mensajes (textos) en profundidad, como por ejemplo, realizar el procesamiento semántico tipo, digamos, *textual entailment* [5].

Cabe mencionar que el software desarrollado funciona utilizando tres capas respecto a la comunicación con el robot. De hecho es un diseño relativamente estándar. Existe una capa de bajo nivel que se encarga de la interacción física con el robot móvil. La programación de esta capa obviamente depende del tipo del robot que estamos utilizando. Otra capa es un programa servidor, que cumple con las funciones de enlace entre el cliente y la capa de bajo nivel. En este sentido, el programa servidor recibe las instrucciones de cliente en un nivel bastante general, por ejemplo, “dar vuelta a la derecha”, o “pararse”, y los codifica (transforma) en los comandos de nivel bajo del robot. El programa cliente se encarga del diálogo con el usuario, presentándole la posición actual del robot en el mapa, mostrándole el progreso de movimientos del robot, presentando las intenciones de los movimientos del robot, procesando los comandos del usuario (realizando tanto el reconocimiento de voz como el análisis lingüístico), generando las respuestas y contestando al usuario (síntesis de voz).

Desde el punto de vista técnico más detallado, la implementación incluye varios submódulos.

1. Desarrollo del parte de servidor que corre en el robot² y recibe los comandos del cliente y los transmite al robot móvil físicamente.
2. Desarrollo del parte de cliente que genere los comandos para el robot móvil y los transmite al servidor.
3. Desarrollo del parte de cliente: desarrollo de la interfaz del usuario para visualizar la interacción con el robot.
4. Desarrollo del parte de cliente: integración del módulo de reconocimiento de voz (librerías de *Dragon SDK*).
5. Desarrollo del parte de servidor y/o cliente: integración del módulo de síntesis de voz (librerías de *Dragon SDK*).
6. Desarrollo del parte de cliente: desarrollo del módulo de análisis lingüístico de las frases de entrada (análisis morfológico y/o sintáctico).
7. Desarrollo del parte de servidor y/o cliente: desarrollo del módulo de generación de las respuestas que corresponden a la situación.
8. Desarrollo del parte de cliente: desarrollo del módulo de estrategias de diálogo, es decir, basándose en la frase de entrada, qué tipo de frase de salida se debe producir y qué acción se debe tomar.

Dentro de cada módulo debemos mencionar las siguientes funciones de los submódulos que se desarrollaron:

1. Desarrollo del parte de servidor que corre en el robot y recibe los comandos del cliente y los transmite al robot móvil físicamente.

² Como ya mencionamos, el robot en este contexto es la computadora que controla al robot.

- Investigación de las opciones posibles disponibles en librerías de *MobileRobots* SDK para la interacción con el robot.
 - Desarrollo de la interfaz del servidor.
 - Desarrollo de la parte que corresponde a la interacción usando el protocolo TCP IP.
2. Desarrollo del parte de cliente que genere los comandos para el robot móvil y los transmite al servidor.
 - Desarrollo de la parte que corresponde a la interacción usando el protocolo TCP IP.
 - Determinación de los comandos que pueden ser interpretados por el servidor.
 3. Desarrollo del parte de cliente: desarrollo de la interfaz del usuario para visualizar la interacción con el robot.
 - Desarrollo de los requisitos de la información requerida por el usuario,
 - Determinación de los parámetros importantes para el comportamiento del robot (robot tiene más de 100 parámetros que cambian su comportamiento, por ejemplo, que tanta gente puede estar en su espacio para que no se considere perdido, velocidad máxima permitida, etc.).
 - Elaboración de la lista de valores recomendados de los parámetros, con opciones de varios entornos.
 - Desarrollo de la representación de los parámetros en la interfaz.
 - Desarrollo de la representación del robot y de su ambiente (el mapa).
 - Desarrollo del manual de elaboración del mapa de ambiente (que es una parte indispensable para el funcionamiento del robot).
 4. Desarrollo del parte de cliente: integración del módulo de reconocimiento de voz (librerías de *Dragon* SDK).
 - Desarrollo de una aplicación de prueba.
 - Desarrollo de un manual de entrenamiento del sistema de reconocimiento (tiene duración alrededor de 5 minutos).
 - Investigación del funcionamiento del sistema sin entrenamiento.
 - Integración de los parámetros del sistema en la interfaz del usuario.
 5. Desarrollo del parte de servidor y/o cliente: integración del módulo de síntesis de voz (librerías de *Dragon* SDK).
 - Desarrollo de una aplicación de prueba.
 - Integración del módulo en la interfaz del usuario.
 - Investigación de la posibilidad de utilizar el apoyo visual al momento del habla (una cabeza parlante en la pantalla).
 6. Desarrollo del parte de cliente: desarrollo del módulo de análisis lingüístico de las frases de entrada (análisis morfológico y/o sintáctico).

- Desarrollo del módulo de análisis morfológico de frase de entrada.
 - Incorporación del análisis sintáctico a través de módulo de análisis sintáctico de *Freeling* [19].
7. Desarrollo del parte de servidor y/o cliente: desarrollo del módulo de generación de las respuestas que corresponden a la situación.
 - Desarrollo de patrones de generación de las respuestas.
 - Desarrollo del módulo de la generación morfológica para los patrones.
 8. Desarrollo del parte de cliente: desarrollo del módulo de estrategias de diálogo, es decir, basándose en la frase de entrada, qué tipo de frase de salida se debe producir y qué acción se debe tomar (gestor de diálogo).
 - Desarrollo de las correspondencias entre las estructuras de frases, su contenido léxico y las situaciones.
 - Detección de las palabras que son marcadores de algunas situaciones.
 - Desarrollo de las estrategias de diálogo, es decir, basándose en la frase de entrada, qué tipo de frase de salida se debe producir y qué acción se debe tomar.

5. Ejemplo de patrones

Vamos a ver un ejemplo. Como una aproximación podemos fijar los siguientes actos de comunicación para los casos más simples:

- Informar_interrupción,
- Informar_estado,
- Informar_falla_de_proceso,
- Informar_exito,
- Informar_falla_de_comunicación

Los actos de comunicación corresponden a las respuestas que se generan pro el robot y se determinan por su estado interno. Se puede considerarlos como unas funciones dentro de las cuales se realiza la sub-respuesta (una respuesta que forma parte de la otra más compleja).

Después de realizar el análisis sintáctico, se determina el verbo principal de la oración. Se determinan los verbos que corresponde a los movimientos: *moverse*, *cambiarse*, *ir*, *llegar*, *avanzar*, *desplazarse*, etc. De esa manera:

VERBO =el verbo principal de la oración,

NP_{meta} = una frase nominal que se encuentra en el mapa y corresponde a una meta.

Podemos utilizar el siguiente metalenguaje para describir la comunicación.

IF (VERBO en ‘verbos de movimiento’ Y NP en METAS)

Grigori Sidorov

```
// (VERBO está en la lista de ‘verbos de movimiento’ Y NP en las METAS del
mapa actual)
THEN
1) INFORMAR_INTERRUPCIÓN (META_ACTUAL)
2) EXITO = PLANEAR_RUTA (META_NUEVA)
3) IF EXITO
    THEN
        4) INFORMAR_EXITO (planear, META_NUEVA)
        5) INFORMAR_ESTADO (META_NUEVA)
    ELSEIF
        6) INFORMAR_FALLA ()
ENDIF
ENDIF
```

Por el momento no proporcionamos la descripción detallada del lenguaje, pero creemos que para cualquier persona que tiene conocimientos básicos de programación es bastante claro lo que sucede dentro de este pseudocódigo.

6. Conclusiones

En este artículo hemos presentado las tareas de desarrollo para el diálogo en lenguaje natural con un robot móvil. Para que este diálogo con el robot sea coherente, aunque fuera sobre un tema específico (en este caso, sobre los movimientos del robot), se necesita que el robot reconozca la frase que se le manda (la voz), la procese, prepare la respuesta según las estrategias de diálogo específicas, genere la frase de respuesta, la pronuncie, y realice la acción solicitada.

Este proyecto desarrolla todas las fases de esta comunicación, integrando el software de reconocimiento y síntesis de voz (*Dragon Naturally Speaking SDK*), el software de manejo de un robot móvil en la arquitectura cliente - servidor, y el software de análisis lingüístico, mismo que el software que implemente un modelo de estrategias de diálogo. Las estrategias de diálogo consisten en varios patrones que se obtienen después de la aplicación del análisis lingüístico de las frases de entrada basándose en la detección de las palabras específicas. El software es aplicable al robot móvil tipo *Pioneer* (que es un estándar de facto en el campo de la robótica móvil de los robots que no son de juguete), aunque sin mayores cambios el software puede ser aplicado a otros tipos de robots.

Agradecimientos. Trabajo realizado con el apoyo de gobierno de la Ciudad de México (proyecto ICYT-DF PICCO10-120), el apoyo parcial del gobierno de México (proyectos CONACYT 50206-H y 83270, SNI) e Instituto Politécnico Nacional, México (proyectos SIP 20111146, 20113295, 20120418, COFAA).

Referencias

1. Gelbukh, A., Sidorov, G.: Procesamiento automático del español con enfoque en recursos léxicos grandes. IPN, 307 p. (2010)
2. Dialog with Robots. AAAI 2010. Fall Symposium. November 2010, Arlington VA. Available: http://hci.cs.wisc.edu/aaai10/?page_id=169 (2010)
3. Lemaignan, S. and Ros, R. and Alami R.: Dialogue in situated environments: A symbolic approach to perspective-aware grounding, clarification and reasoning for robot. In: Proc. Robotics, Science and Systems, Grounding Human-Robot Dialog for Spatial Tasks workshop (2011)
4. Torres Goterris, F.: Sistemas de diálogo basados en modelos estocásticos. PhD Thesis, UPV, Valencia (2006)
5. Pakray, P., Barman, U., Bandyopadhyay, S., Gelbukh, A.: A Statistics-Based Semantic Textual Entailment System.. MICAI 2011, Lecture Notes in Artificial Intelligence N 7094, Springer, pp. 267–276 (2011)
6. Levin, E., Pieraccini, R., Eckert, W., Di Fabbrizio, G., Narayanan, S.: Spoken language dialogue: from theory to practice. In: Proc. of ASRU–IEEE Workshop. Keystone, Colorado, USA (1999)
7. Levin, E., Pieraccini, R., Eckert, W.: A stochastic model of human-machine interaction for learning dialog strategies. IEEE Transactions on Speech and Audio Processing. vol. 8 (1), pp. 11–23 (2000)
8. Marge, M., Pappu, A., Frisch, B., Harris, Th. K., Rudnicky, A.: Exploring Spoken Dialog Interaction in Human-Robot Teams. In: Proceedings of Robots, Games, and Research: Success stories in USARSim IROS Workshop, October 2009, St. Louis, MO, USA (2009)
9. Roy, D., Reiter, E.: Connecting language to the world. Artificial Intelligence (2005)
10. Sisbot, E.A., Ros, R., Alami, R.: Situation assessment for human-robot interaction. In: Proc. of 20th IEEE International Symposium in Robot and Human Interactive Communication (2011)
11. Kruijff, G.J.M., Lison, P., Benjamin, T., Jacobsson, H., Zender, H., Kruijff-Korbayova, I., Hawes, N. Situated dialogue processing for human-robot interaction. Cognitive Systems, pp. 311–364 (2010)
12. Searle, J.: Indirect speech acts. In: Syntax and Semantics, 3: Speech Acts, ed. P. Cole & J. L. Morgan, pp. 59–82. New York: Academic Press (1975) [Reprinted in Pragmatics: A Reader, ed. S. Davis, pp. 265–277. Oxford: Oxford University Press (1991)]
13. Grice, P.: Logic and conversation. In: Syntax and Semantics, 3: Speech Acts, ed. P. Cole & J. Morgan, New York: Academic Press (1975) [Reprinted in Studies in the Way of Words, ed. H. P. Grice, pp. 22–40. Cambridge, MA: Harvard University Press (1989)]
14. Souvignier, B., Kellner, A., Rueber, B., Schramm, H., Seide, F.: The Thoughtful Elephant: Strategies for Spoken Dialog Systems. IEEE Transactions on Speech and Audio Processing. vol. 8 (1), pp. 51–62 (2000)

15. Pineda, L.A., Estrada, V.M., Coria, S.R., Allen, J.F.: The obligations and common ground structure of practical dialogues. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 11(36), pp. 9-17 (2007)
16. Levinson, S.C. *Pragmatics*, Cambridge University Press, Cambridge, MA, USA (1983)
17. Allen, J.F., Core, M.G.: DAMSL: Dialog Act Markup in Several Layers (Draft 2.1), Multiparty Discourse Group, Discourse Research Initiative (1997)
18. Jurafsky, D., Martin, J.: *Speech and Language Processing*. Prentice Hall (2009)
19. Padró, L., Collado, M., Reese, S., Lloberes, M., Castellón, I.: FreeLing 2.1: Five Years of Open-Source Language Processing Tools. In: *Proceedings of 7th Language Resources and Evaluation Conference (LREC 2010)*, ELRA La Valletta, Malta (2010)

Clasificación de servicios Web empleando estrategias supervisadas

Víctor O. Díaz-Torres, Esaú Villatoro-Tello,
Christian Sánchez-Sánchez y Héctor Jiménez-Salazar

Departamento de Tecnologías de la Información,
División de Ciencias de la Comunicación y Diseño,
Universidad Autónoma Metropolitana Unidad Cuajimalpa, México DF
{209363180, evillatoro, csanchez, hjimenez}@correo.cua.uam.mx

Resumen El incremento de Servicios Web disponibles en catálogos públicos en Internet ha provocado la urgente necesidad de proponer nuevas formas de búsqueda y categorización de los mismos. El lenguaje de descripción WSDL permite a los proveedores de Servicios Web dar detalles del funcionamiento de un servicio al momento de registrarlo en algún catálogo. El problema principal de esta técnica es que se depende en gran medida de la experiencia y criterio del proveedor al momento de asignar una categoría. Dentro de este trabajo se propone una nueva forma de representación de los Servicios Web que aprovecha la información contenida en los atributos de descripción contenidos en el WSDL. Nuestra propuesta emplea técnicas tradicionales de clasificación de textos basadas en aprendizaje supervisado. Los resultados obtenidos muestran que es posible obtener resultados eficientes en la clasificación aún sin contar con el atributo *Documentation* de los Servicios Web.

Palabras clave: servicios web, documentos WSDL, clasificación de textos, aprendizaje supervisado, procesamiento de lenguaje natural.

1. Introducción

Actualmente, en el ámbito del desarrollo de Software existe una fuerte motivación para preservar y practicar las *buenas costumbres de programación*. Entre las ventajas que estas buenas costumbres de programación ofrecen están: *i)* alto rendimiento de aplicaciones al permitir su funcionamiento de manera distribuida, *ii)* colaboración a través de mecanismos estandarizados, *iii)* reutilización (reuso) de software, y, *iv)* reducción de costos en los procesos de desarrollo de software.

Los Servicios Web (WS, por sus siglas en Inglés) son un ejemplo de los mecanismos que surgieron a partir del conjunto de necesidades mencionadas en el párrafo anterior. Agregado a esto, los WS surgieron como una tendencia de negocios sobre aplicaciones, que en el fondo contienen descripciones encapsuladas de una aplicación (métodos y/o funciones) que son de uso público y que están

basados en estándares de Internet, tales como lo son los lenguajes WSDL¹, OWL-S o WSMO.

En años recientes el número WS ha crecido considerablemente, lo cual dificulta la tarea de buscar y localizar efectivamente un WS. Agregado a esto, para la mayoría de las personas no es fácil leer e interpretar la información (descripción y mensajes para su ejecución) referente a los WS. Con la finalidad de apoyar en la solución a este problema han surgido registros públicos en los cuales se propone un estándar a seguir cada vez que se produce un nuevo WS. El problema con este tipo de servicios es que dependen en gran manera del criterio y experiencia del proveedor al momento de registrar su WS.

Con la finalidad de poder dar soluciones efectivas a los problemas antes mencionados, en este trabajo se propone una técnica de clasificación automática de WS que aprovecha los componentes de la descripción WSDL de un servicio Web. WSDL (Web Service Description Language) es una de las formas más comúnmente empleadas para definir los servicios web. Una descripción en WSDL aprovecha las ventajas ofrecidas por la gramática de XML para definir los medios de comunicación, el intercambio de mensajes y las reglas de operación de un WS. Dentro de un documento WSDL, la definición abstracta de las operaciones y de los mensajes se encuentran separadas de su implementación concreta, lo que permite la reutilización de las definiciones abstractas: mensajes, los cuales son descripciones abstractas de los datos que son intercambiados, y de los tipos de puertos los cuales son colecciones abstractas de las operaciones.

Nuestros experimentos muestran que a través de utilizar ciertos componentes de la descripción WSDL es posible alcanzar resultados prometedores al momento de realizar la clasificación de servicios Web. Nuestra propuesta emplea técnicas tradicionales de Procesamiento de Lenguaje Natural para representar los documentos WSDL y además, por medio de una estrategia de aprendizaje supervisado se construye un modelo de clasificación de WS.

El resto del documento se encuentra organizado de la siguiente manera. En la sección 2 se describe brevemente el trabajo relacionado más relevante a la temática en cuestión. En la sección 3 se describe el método propuesto así como las técnicas empleadas. La sección 4 detalla los conjuntos de datos, la configuración experimental y los resultados obtenidos. Finalmente, la sección 5 muestra las conclusiones obtenidas y define las líneas de trabajo futuro.

2. Trabajo relacionado

En [3] se propone un enfoque para clasificar y anotar semánticamente Servicios Web. La clasificación de los servicios a un dominio específico lo realizan usando Maquinas de Soporte Vectorial (MSV), posteriormente identifican conceptos clave en la documentación de los servicios, los cuales son relacionados por medio de un enrejado de conceptos para anotar los servicios. Los autores reali-

¹ Algunos ejemplos catálogos de WS disponibles a través de UDDIs (Universal Description, Discovery and Integration) en Internet son Xmethods [1] y Seekda [2].

zaron pruebas con un conjunto de 205 Servicios Web distribuidos en 11 clases², y obteniendo una precisión en la clasificación de 63 %.

Un trabajo similar es fue el realizado por [4], donde combinaron la clasificación de documentos y la alineación de ontologías para enriquecer semánticamente las descripciones de los Servicios Web. Los autores clasifican los servicios a partir de calcular el centroide de cada categoría, para que posteriormente cuando llegue un nuevo documento este sea incluido a la categoría que tenga el centroide más cercano. Posteriormente, relacionan la información en la descripción del servicio con conceptos en una ontología y ya teniendo los servicios anotados, entonces proceden a buscar conceptos similares en otras ontologías. Las pruebas las realizaron con un conjunto de 391 Servicios Web³ tomando un subconjunto de 100 descripciones de servicio que anotaron semánticamente de manera manual obteniendo un total de 6 categorías. Los resultados que ellos reportan son con respecto a la precisión del anotado, de acuerdo a una métrica que proponen, donde obtuvieron un promedio de la precisión de la anotación del 85.7%, ya que reportan que el anotado semántico ayuda a corregir los errores del modelo de clasificación.

Otra forma de clasificar Servicios Web, fue la propuesta en [5], en la cual a partir de servicios anotados semánticamente proponen una serie de heurísticas para determinar a qué categoría pertenecen los Servicios Web, ellos proponen comparar el servicio con todos los servicios de cada categoría verificando a cuales es más parecido, en la comparación verifican la similitud de las conceptos (en ontologías) relacionados con las operaciones del servicio y de sus entradas y salidas. Los autores utilizaron un corpus de 164 descripciones de Servicios Web (anotados semánticamente) pertenecientes a 23 diferentes categorías⁴. Ellos obtuvieron un promedio de éxito en la clasificación de 83 %, también debida a la anotación semántica.

En [6] clasifican por medio de máquinas de soporte vectorial, los Servicios Web de acuerdo a la categoría UNSPSC (United Nations Standard Products and Services Code). En este trabajo ellos toman las características funcionales del servicio (nombre de operaciones, entradas y salidas). Para hacer la clasificación, y debido a que UNSPC es una jerarquía de categorías, ellos emplean primero clasificadores que distinguen las categorías padre y posteriormente subclasificadores que ayudan a determinar la categoría dado ese padre. También obtienen términos similares por medio de WordNet, ya que las MSV pueden trabajar con vectores con muchos términos. Los autores trabajaron con una colección de 1007 Servicios Web anotados semánticamente⁵, preclasificados en siete categorías. Hicieron dos tipos de experimentos: en el primero utilizaron tres diferentes medidas de similitud semántica en dos diferentes funciones *kernel* de MSV, el mejor resultado lo obtuvieron con la medida de similitud Path Length y RBF de función kernel con el 90.5 % de exactitud. Y en los segundos experi-

² Disponibles en <http://moguntia.ucd.ie/repository/ws2003>

³ Disponibles en <http://www.andreas-hess.info/projects/annotator/>

⁴ Disponibles en <http://www.andreas-hess.info/projects/annotator/>

⁵ Descargados de <http://projects.semwebcentral.org/projects/owlstc/>

mentos solo seleccionando 500 Servicios Web utilizando la medida de similitud y el kernel que les dio mejores resultados, compararon con otros métodos de clasificación (MSV con información semántica y textual como características de clasificación y método de comparación propiedad-valor) su método ganó con el 90.5 % mostrado previamente.

El enfoque propuesto dentro de este trabajo se diferencia del trabajo previo principalmente en que no utiliza fuentes externas de conocimiento para llevar a cabo la representación de las descripciones de los servicios Web. Agregado a esto se propone un modelo más realista, pues los Servicios Web que se encuentran en Internet son muy heterogéneos y no siempre cuentan con toda la información de sus componentes. Por ello nos orientamos a seleccionar combinaciones de características extraídas de los Servicios Web para validar cuales de estas nos ayudan a una mejor y menos costosa clasificación.

3. Método propuesto

3.1. Pre-procesamiento de los documentos WSDL

El pre-procesamiento consiste en la extracción de las palabras contenidas en los atributos de la descripción de los WS. Los atributos de descripción considerados para nuestros experimentos fueron: *nombre del servicio*, *documentación*, *nombre de los mensajes* y *nombre de los parámetros*. Es importante mencionar que de todos estos atributos, el único que se encuentra en lenguaje natural es el de *documentación*, por lo tanto su pre-procesamiento es relativamente sencillo. Por el contrario, para el resto de los atributos (*i.e.*, *nombre del servicio*, *mensajes* y *parámetros*) fue necesario definir un conjunto de heurísticas más elaboradas para poder extraer elementos relevantes para nuestros fines. Para la realización de nuestros experimentos se aplicaron el mismo conjunto de reglas definidas en [8], las cuales son:

- Filtrado de términos. Utilizando un diccionario, definido a través de *Wordnet*, se filtran palabras desconocidas de los atributos. El objetivo es conservar sólo términos conocidos para la representación de los WS.
- División de términos compuestos. Para esto se emplean las técnicas definidas en [8], donde el objetivo es descubrir términos adicionales que posiblemente pueden aportar información valiosa a la representación del WS.
- Eliminación de palabras vacías. Las palabras vacías representan palabras carentes de información tales como: artículos, preposiciones, pronombres, etc. Al eliminar este tipo de palabras es posible reducir la cantidad de *ruido* en la representación de los WS.

3.2. Representación de los WS

Como se ha venido mencionando en secciones previas, atacamos el problema de la clasificación de servicios web empleando el paradigma de clasificación de

textos (CT)⁶. Bajo este paradigma un primer paso necesario es el *indexado* de los documentos de entrenamiento (Tr), actividad que corresponde al mapeo de un documento d_j en una forma compacta de su contenido. La representación más comúnmente utilizada para representar cada documento es un vector con términos ponderados como entradas, concepto tomado del modelo de espacio vectorial usado en recuperación de información [7]. Es decir, un texto d_j es representado como el vector $\vec{d}_j = \langle w_{kj}, \dots, w_{|\tau|j} \rangle$, donde τ es el *diccionario*, *i.e.*, el conjunto de términos que ocurren al menos una vez en algún documento de Tr , mientras que w_{kj} representa la importancia del término t_k dentro del contenido del documento d_j . En ocasiones τ es el resultado de filtrar las palabras del vocabulario, *i.e.*, resultado de un preprocesamiento (sección 3.1). Una vez que hemos hecho los filtrados necesarios, el diccionario τ puede definirse de acuerdo a diferentes criterios, sin embargo el que se empleó en esta propuesta corresponde a la Bolsa de Palabras.

La Bolsa de Palabras (BOW)⁷ es la forma tradicionalmente utilizada para representar los documentos [9]. Este método de representación utiliza a las palabras simples como los elementos del vector de términos.

Con respecto al peso (*i.e.*, la importancia) w_{kj} , se tienen diferentes formas de calcularlo, entre las más usadas en la comunidad científica se tienen el ponderado booleano, ponderado por frecuencia de término y el ponderado por frecuencia relativa de términos. Una breve descripción es dada a continuación:

- *Ponderado Booleano*: Consiste en asignar el peso de 1 si la palabra ocurre en el documento y 0 en otro caso.

$$w_{kj} = \begin{cases} 1, & \text{si } t_k \in d_j \\ 0, & \text{en otro caso} \end{cases} \quad (1)$$

- *Ponderado por frecuencia de termino (TF)*: En este caso el valor asignado es el número de veces que el término t_k ocurre en el documento d_j .

$$w_{kj} = f_{kj} \quad (2)$$

- *Ponderado por frecuencia relativa (TF-IDF)*: Este tipo de ponderado es una variación del tipo anterior y se calcula de la siguiente forma:

$$w_{kj} = TF(t_k) \times IDF(t_k) \quad (3)$$

donde $TF(t_k) = f_{kj}$, es decir, la frecuencia del termino t_k en el documento d_j . IDF es conocido como la “frecuencia inversa” del termino t_k dentro del documento d_j . El valor de IDF es una manera de medir la “rareza” del termino t_k . Para calcular el valor de IDF se utiliza la siguiente formula:

$$IDF(t_k) = \log \frac{|D|}{\{d_j \in D : t_k \in d_j\}} \quad (4)$$

donde D es la colección de documentos que está siendo indexada.

⁶ La Clasificación de Textos es la tarea de asociar automáticamente categorías predefinidas con documentos a partir del análisis de su contenido [9].

⁷ En inglés se dice “Bag of Words”.

3.3. Algoritmo de aprendizaje

En la literatura referente a la CT [9] existe gran variedad de algoritmos que han sido evaluados y que han mostrado ser apropiados para la tarea de clasificación de textos. Dado que el objetivo de este trabajo no es evaluar exhaustivamente cuál podría ser el mejor clasificador o aprendiz, si no más bien evaluar la pertinencia de la representación propuesta; mostramos resultados solo al trabajar con el algoritmo de *k-means*⁸ [10].

El algoritmo *k-means* calcula en un primer paso un valor *media* o *centroide* m_i para cada una de las clases. Cuando se proporciona un nuevo elemento sin clasificar, este es asignado a la clase i con cuyo centroide m_i se tenga la menor distancia, *i.e.*, la mayor similitud. Para nuestros experimentos, el centroide de cada una de las categorías de WS es definido siguiendo una forma de representación vectorial como se explico en la sección 3.2.

Para el calculo de similitud se han propuesto varias métricas que permiten determinar el parecido de pares de documentos [11]. El objetivo de estas métricas es contar con un valor numérico al cual llamaremos coeficiente de similitud SC , el cual nos dirá cuán parecidos son los documentos D_i y D_j , note que cualquiera de estos dos documentos puede representar el centroide m_i . Dos medidas ampliamente utilizadas en el campo de recuperación de información que permiten determinar la similitud entre documentos son:

- **Medida Cosenoidal.** La idea básica de ésta es medir el ángulo entre el vector de D_i y de D_j , para hacerlo, calculamos:

$$SC(D_i, D_j) = \frac{\sum_{k=1}^t w_{ik}w_{jk}}{\sqrt{\sum_{k=1}^t (w_{jk})^2 \sum_{k=1}^t (w_{ik})^2}} \quad (5)$$

- **Medida DICE.** El coeficiente de DICE es obtenido por medio de:

$$SC(D_i, D_j) = \frac{2 \sum_{k=1}^t w_{ik}w_{jk}}{\sum_{k=1}^t (w_{jk})^2 + \sum_{k=1}^t (w_{ik})^2} \quad (6)$$

En todos los casos k va de 1 a el número total de términos del vocabulario τ , w_{ik} indica la importancia del término k en el documento D_i mientras que w_{jk} la importancia del término k en el documento D_j .

3.4. Evaluación

Para evaluar un sistema de clasificación de textos se utilizan las medidas de *Precisión* y *Recuerdo*, que son medidas comunes en el área de recuperación de información. La precisión (P) es la proporción de documentos clasificados correctamente en una clase c_i con respecto a la cantidad de documentos clasificados en esa misma clase. El recuerdo (R), la proporción de documentos clasificados

⁸ El algoritmo *k-means* es también conocido como *k-medias*

correctamente en una clase c_i con respecto a la cantidad de documentos que realmente pertenecen a esa clase. Así, la precisión se puede ver como una medida de la corrección del sistema, mientras que el recuerdo da una medida de cobertura o completitud.

Normalmente se emplea la medida F para describir el comportamiento de la clasificación, la cual se define como:

$$F = \frac{(1 + \beta^2)Precision * Recuerdo}{\beta^2 Precision + Recuerdo} \quad (7)$$

donde con $\beta = 1$ representa la media armónica entre la precisión y el recuerdo. La función de β es la de controlar la importancia relativa entre las medidas de precisión y recuerdo. Es común asignar un valor de 1 indicando igual importancia a ambas medidas.

4. Experimentos y resultados

4.1. Conjunto de datos

Para nuestros experimento se trabajo con la colección de ASSAM⁹. La colección fue filtrada aplicaron las mimas técnicas que se describen en [8], resultando en un total de 203 documentos WSDL repartidos en 22 clases.

4.2. Definición experimentos

Un conjunto de 7 configuraciones de experimentos fueron propuestos, los cuales consistieron en determinar el aporte de información de cada uno de los atributos extraídos de los documentos WSDL. Como se mencionó en la sección 3.1, los atributos de descripción considerados fueron: *documentación (Doc)*, *nombre servicio (Nom)*, *mensajes (Msgs)* y *parámetros (Param)*. La configuración base de nuestros experimentos esta dada por el uso únicamente del atributo *Doc*; idealmente todo proveedor de un WS deberá proporcionar una documentación lo suficientemente clara que permita hacer una clasificación correcta del mismo.

Agregado a esto se consideró las combinaciones de atributos siguientes: *Nom + Msgs*, *Nom+Param* y *Nom+Msgs+Param*, donde la finalidad de estos experimentos fue demostrar que es posible determinar la categoría de un WS sin la necesidad de contar con una *documentación*. La tabla 1 muestra el tamaño del vector del vocabulario para cada uno de los experimentos propuestos.

4.3. Resultados

La gráfica mostrada en la figura 1 muestra los resultados obtenidos por el clasificador *k-means* al emplear como medida de similitud el coeficiente DICE.

⁹ <http://www.andreas-hess.info/projects/annotator/>

Tabla 1. Tamaño del vector de características correspondiente a cada uno de los atributos de descripción considerados en los experimentos.

Experimento	Tam. Vocabulario
Doc	2599
Nomb	748
Param	799
Msgs	656
Nomb+Msgs	755
Nomb+Param	930
Nomb+Msgs+Param	935

Las gráficas muestran el desempeño del clasificador en términos de la medida F (sección 3.4).

Nótese que los resultados obtenidos muestran que el uso de *Doc* resulta en un desempeño bajo del clasificador, apenas logrando un 0.72 bajo un esquema de pesado TF-IDF. De igual forma los resultados muestran que manejar sólo *Nom*, *Msgs* y/o *Param* no permite mejorar los resultados notoriamente, sin embargo, es importante mencionar que el tamaño del vector de estas configuraciones es de menos de la mitad comparado con el empleado por *Doc* (tabla 1), lo cual indica que contienen información relevante para el clasificador.

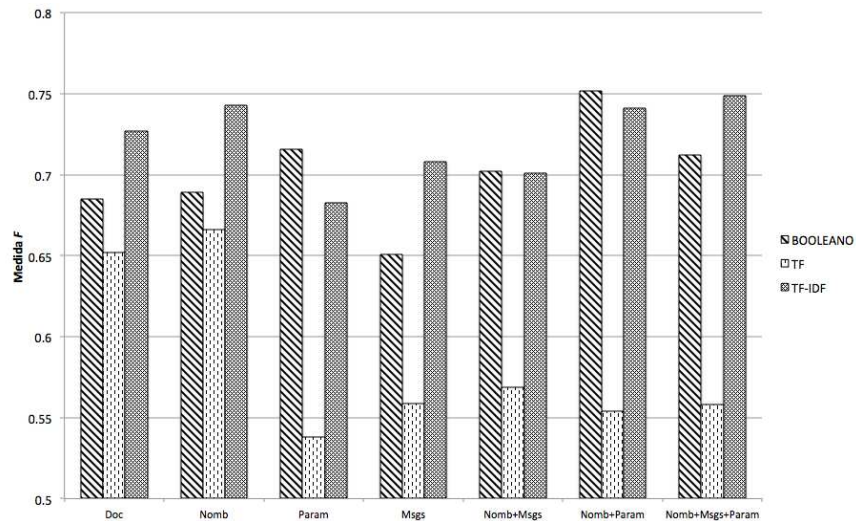


Figura 1. Resultados obtenidos empleando como medida de similitud el coeficiente DICE.

Los mejores resultados que se obtuvieron fue bajo la configuración de *Nom + Param* bajo un esquema de pesado *booleano*, alcanzando un desempeño de 0.75 en la medida F . Otro factor importante que vale la pena resaltar es el hecho

de que manejar un esquema de *presencia* de los atributos, *i.e.*, un esquema de pesado *booleano*, aporta mayor información al clasificador que los esquemas de *frecuencia* y *frecuencia relativa* respectivamente.

La gráfica mostrada en la figura 2 muestra los resultados obtenidos por el clasificador *k-means* al emplear como medida de similitud el Coseno.

Como se puede observar, la configuración *Nom+Param* supera de manera considerable al método base logrando un 0.81 en la medida *F* contra un 0.72 obtenido al usar sólo *Doc*. Nóte también, que el resultado de estos experimentos muestra el mismo comportamiento de los mostrados en la tabla 1, *i.e.*, una forma de representación *booleana* aporta mejores elementos al clasificador que las frecuencias.

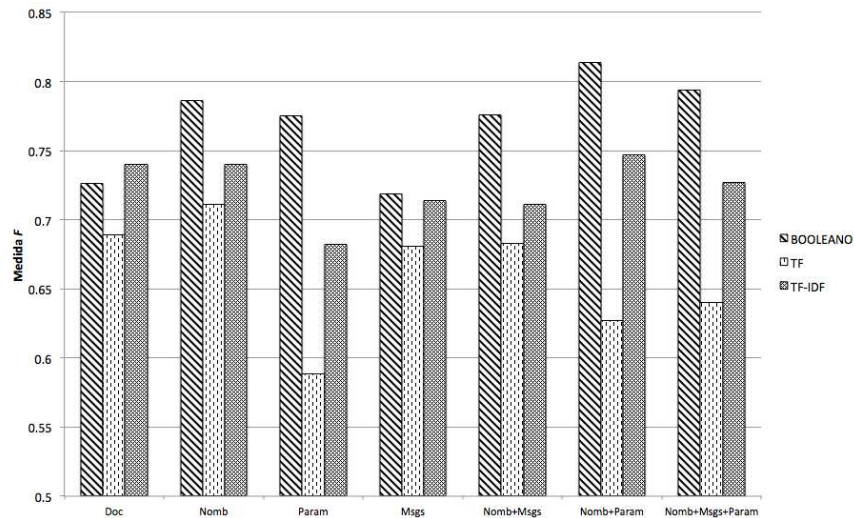


Figura 2. Resultados obtenidos empleando como medida de similitud el Coseno.

Finalmente, los resultados mostrados en la figura 2 muestran que el uso de una medida de similitud más fina (*e.g.*, el Coseno) permite distinguir mejor entre las categorías de los servicios Web.

5. Conclusiones

En este trabajo se ha propuesto una nueva forma de representación y de clasificación de Servicios Web que aprovecha las ventajas de la información contenida en las descripciones WSDL. El método propuesto emplea técnicas de Clasificación de Textos que han sido tradicionalmente empeladas en el área de Procesamiento de Lenguaje Natural.

Se mostró un estudio del aporte que tienen los atributos de descripción contenidos en un WSDL. Los resultados obtenidos muestran que es posible prescindir

de la existencia de la *documentación* y aún así obtener buenos resultados durante la clasificación. Los resultados obtenidos son alentadores, pues nos permiten afrontar la problemática de la ausencia de *documentación* en los documentos WSDL, el cual es un problema muy frecuente.

Cómo trabajo futuro se pretende evaluar el aporte que podrían tener representaciones basadas en *n*-gramas[12]. Los *n*-gramas han mostrado ser efectivos en tareas de clasificación de textos gracias a que permiten capturar información del contexto agregado a que permiten mantener el orden de aparición de las palabras, cosa que no sucede con una representación de BOW. Intuitivamente, el uso de *n*-gramas permitirá tener una representación más fina con la cual podrían mejorarse los resultados de clasificación.

Agradecimientos. Agradecemos a la Universidad Autónoma Metropolitana Unidad Cuajimalpa y al proyecto CONACYT número CB2010/153315 por el apoyo para la asistencia a este evento.

Referencias

1. Xmethods, <http://xmethods.net/ve2/index.po> (Última visita en Abril de 2013)
2. Seekda, <http://webservices.seekda.com/> (Última visita en Noviembre 2012)
3. Bruno, M., Canfora, G., Di Penta, M., Scognamiglio, R. (2005) An Approach to support Web Service Classification and Annotation. En *Proceeding IEEE '05 Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)*. IEEE Computer Society Washington.
4. Crasso, M., Zunino, A., Campo, M. (2010) Combining Document Classification and Ontology Alignment for Semantically Enriching Web. En *New generation computing*. Vol. 28, No. 4, pp. 371-403.
5. Corella, M. A. y Castells, P. (2006) Semi-automatic semantic-based web service classification. En *Proc. of the International Conference on Knowledge-Based Intelligent Information and Engineering Systems*.
6. Wang, H., Shi, Y., Zhou, X., Zhou, Q., Shao, S. y Bouguettaya, A. (2010) Web Service Classification Using Support Vector Machine. En *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. pp.3-6
7. Baeza-Yates, R., y Ribeiro-Neto, B. (1999) *Modern Information Retrieval*, Addison Wesley.
8. Jiménez-Salazar, H., Sánchez-Sánchez, C., Rodríguez-Lucatero, C. y Luna-Ramírez, A. W. (2012) An Analysis of Web Services Attributes for Discovery Support. En *Research In Computing Science*
9. Sebastiani F. (2002) Machine Learning in Automated Text Categorization. En *ACM Computing Surveys*, Vol. 34, No. 1, March 2002, pp. 1-47.
10. Duda, O., Hart, P.E., y Stork, D.G. (2000) En *Pattern Classification*, John Wiley & Sons.
11. Grossman, D. A. y Frieder, O. (2004) En *Information Retrieval, Algorithms and Heuristics*. Springer, second edition edition.
12. Sidorov G., Velasquez F., Stamatatos E., Gelbukh A., Chanona-Hernández L..(2013) Syntactic Dependency-Based N-grams: More Evidence of Usefulness in Classification. CICLing 2013. En *Lecture Notes in Computer Science*. Vol. 7816, pp. 13-24.

Representación basada en grafos para la identificación de autoría en textos para el idioma español

Nahum Loya¹, Iván Olmos¹, David Pinto¹ y Jesús González²

¹Benemérita Universidad Autónoma de Puebla,
Facultad de ciencias de la computación, México

²Instituto de Astrofísica, Óptica y Electronica,
Departamento de ciencias de la computación, México
{nahum.loya,iolmos,dpinto}@cs.buap.mx, jagonzalez@inaoep.mx

Resumen El proceso de identificación de autoría consiste en determinar al autor que ha escrito algún tipo de documento. Para realizarse es necesario conocer las características más adecuadas que permiten identificar los rasgos de escritura. Este trabajo propone una metodología basada en representaciones de textos a través de grafos etiquetados. Los grafos son utilizados como elemento esencial para la búsqueda de patrones o características. Se trabaja con un conjunto de datos que corresponde al conjunto de documentos (libros) pertenecientes a seis autores cada autor con cinco documentos diferentes. Se realiza una fase de preprocesamiento consistente en la eliminación de palabras cerradas y signos de puntuación, posteriormente los documentos son separados en frases. Dichas frases son representadas a través de grafos etiquetados, haciendo uso de dos representaciones propuestas. El objetivo es extraer características importantes. Dichas características son usadas en el proceso de clasificación. Los resultados muestran una mejora tomando como métrica los niveles de precisión y en comparativa con otros métodos como la bolsa de palabras o las características estilométricas.

Palabras clave: atribución de autoría, representaciones de texto en grafos, autoría en español, clasificación.

1. Introducción

La atribución de autoría suele verse como un tema de la lingüística computacional y tiene como objetivo identificar al autor o escritor original de un texto. Para lograrlo es necesario buscar características o perfiles que identifiquen plenamente al autor. Esta no es una tarea trivial debido a que los estilos de escritura suelen ser similares, inclusive los textos suelen ser escritos por diversos autores lo que dificulta aún más el proceso de búsqueda de patrones.

Actualmente se han estudiado diferentes técnicas para solventar el problema, las cuales van desde aplicaciones matemáticas que intentan medir el grado de frecuencia haciendo uso de métricas tradicionales como la varianza y la desviación estándar y modelos de aprendizaje automático.

En lo que compete al ámbito lingüístico computacional se han buscado características que van desde examinar las palabras en el texto, métricas como el tamaño de la letra, el número de vocales o consonantes utilizadas e inclusive se han realizado análisis de errores sintácticos y ortográficos.

El interés en esta área actualmente radica en que existen diversos documentos que no tienen asociado un autor o son considerados anónimos. Por tanto existe una gran necesidad de desarrollar modelos automáticos y confiables para la extracción de características que coadyuven a la solución óptima del problema.

Ahora que se ha establecido la problemática, en este sentido se presenta un modelo de representación de texto mediante grafos. Los grafos son una manera natural de representar la información, dado que se puede establecer conceptos a través de los nodos y relaciones a través de las aristas. Los documentos se pueden dividir en párrafos y estos en enunciados. En estos existen relaciones entre las palabras tal y como los nodos de los grafos son ligados a través de aristas. Hablamos de grafos etiquetados en el sentido de hacer una extensión de la noción típica de grafo (conjunto de vértices y aristas). Esta extensión permite incorporar relaciones habituales en el texto a través de los grafos.

El caso de estudio que se plantea considera 6 autores literarios, cada uno con cinco documentos (libros) diferentes pertenecientes al género de aventura. Se pretende mediante este conjunto de datos evaluar la calidad de la metodología la cual tiene como idea fundamental el uso de los grafos como herramienta de representación.

El resto del artículo se encuentra conformado de la siguiente manera. En la sección 2 se hace una revisión de los trabajos relacionados considerando dos aspectos: la atribución de autoría y el uso de grafos en el area de lingüística computacional. En la sección 3 se presenta una notación que es empleada para comprender el tipo de estructura de grafos propuesta en este trabajo. En la sección 4 se presentan dos propuestas que son usadas para la extracción automática de características, además de establecer una justificación del uso de las representaciones. En la sección 5 se detalla la metodología usada en este trabajo así como se presenta el caso de estudio. En la sección 6 se presenta un análisis de resultados, así como una comparativa entre los resultados obtenidos con los algoritmos de clasificación y un contraste entre los modelos de grafos y otros similares. Finalmente en la sección 7 se presentan las conclusiones y el trabajo a futuro.

2. Trabajo previo

El trabajo de atribución de autoría se plantea como un proceso típico de clasificación donde es necesario extraer características y las clases son los diferentes autores. Los primeros modelos para solventar el problema plantean el uso de bolsa de palabras como conjunto de características; en este caso se considera cada palabra del vocabulario como un atributo [6,13]. La desventaja radica en que estos modelos no muestran las relaciones entre el texto, enunciados o palabras. Otros enfoques similares tratan de extraer características a partir de las

palabras (total de palabras, tamaño de las palabras, número de vocales, número de consonantes, signos de puntuación, etc.), que al final son representadas como vectores de soporte [10]. Enfoques minuciosos son los relacionados con el análisis de las partes de la oración (adverbios, verbos, adjetivos, etc.) [8]. Estos incorporan información relacionada con la estructura del lenguaje empleado como lo son elementos léxicos, sintácticos [7]. Un método frecuente es el uso de n-gramas de palabras o de caracteres, donde se busca encontrar la estructura de la escritura [9]. Un trabajo sobresaliente es el mostrado en [5], donde se hace la evaluación de 39 modelos o procedimientos para caracterizar textos, dicha evaluación es realizada sobre un corpus formado por textos de una serie de columnistas del diario “Telegraphe” de Londres. Los resultados obtenidos muestran el porcentaje de confiabilidad de distintos métodos (palabras y marcas de puntuación, bigramas y trigramas de palabras, etc.) en comparación con la cantidad de autores (clases). Como resultado de este estudio se observa que los métodos obtienen una confiabilidad de hasta el 88% cuando se consideran dos autores y baja radicalmente hasta 34% cuando se trabaja con más de 10 autores.

En lo que concierne al tema de uso de grafos en el procesamiento del lenguaje natural, sin lugar a dudas son una buena forma de representación de dominios textuales. En la literatura distintas investigaciones relacionadas con la minería textual proponen el uso de los grafos. Desde la década de los 80's los grafos han sido usados para representar relaciones y conceptos. Basta con citar trabajos como los de [12], pionero en proponer los grafos conceptuales como un modelo para representar conocimiento. Estos grafos son concebidos de una forma psicológica, lingüística.

En particular en lo que concierne a la aplicación de determinación de autoría, los grafos también han sido utilizados con el objetivo de extraer características determinantes para la detección del autor. Ejemplo de este tipo de aplicaciones se muestran en [2], quien a partir de la representación de grafos de escritura logra obtener características importantes para la detección de autoría.

Otro estudio es el desarrollado en [11], donde se hace uso de parser de dependencias de Stanford para construir árboles que representan las relaciones sintácticas entre palabras denominadas “sn-gramas”. Se muestra que la diferencia entre los n-gramas tradicionales y los sn-gramas radica en la forma en como son considerados los vecinos. Los “sn-gramas” son aplicados para extraer características del texto para atribución de autoría de tres autores logrando resultados favorables de hasta el 100%.

Como se puede observar, en este análisis realizado existen diferentes propuestas para representar dominios textuales, con ventajas y desventajas. En este sentido, este trabajo de investigación explora una alternativa que permita abordar la representación de información en textos a través de grafos tomando como caso de estudio la atribución de autoría.

En la siguiente sección se presentan las nociones teóricas para la representación de grafos etiquetados. Se realiza una extensión de la notación tradicional de grafos con el objetivo de representar grafos etiquetados.

3. Extracción de características y representación basada en grafos

Tomando como base la noción de grafo, los grafos son representados por una dupla de la forma $G = (V, E)$, donde V es el conjunto de vértices del grafo y $E \subseteq V \times V$ representa la asociación entre vértices mediante arcos. Sin embargo, esta notación no es útil para el caso de representación de conocimiento, ya que no solamente es de nuestro interés guardar la estructura del grafo sino además el contenido de la información que se quiera mapear (etiquetas a los vértices y a los arcos), por ello introducimos la siguiente notación.

Definición 1 *Grafo.* Es una sextupla donde: $G = (V, E, L_V, L_E, \alpha, \beta)$

- $V = \{v_i | i = 1, \dots, n\}$ es un conjunto finito de vértices, $V \neq \emptyset$, y $n = \#vertices$ en el grafo.
- $E \subseteq V \times V$ es el conjunto finito de aristas, $E = \{e = \{v_i, v_j\} | v_i, v_j \in V, 1 \leq i, j \leq n\}$.
- L_V , es un conjunto de etiquetas para los vértices.
- L_E , es un conjunto de etiquetas para las aristas.
- $\alpha : V \rightarrow L_V$, es una función que asigna las etiquetas a los vértices.
- $\beta : E \rightarrow L_E$, es una función que asigna etiquetas a las aristas.

En esta representación se asume que no existe una dirección entre el vértice origen (v_o) y el vértice destino (v_d) que unen cada arco. Si se quisiera representar un grafo dirigido se puede utilizar la notación para un arco $\{e = (v_o, v_d)\}$ (en lugar de utilizar llaves), indicando que los paréntesis establecen un orden.

Una vez introducida la notación en la subsecuente sección se presentan dos propuestas basadas en grafos etiquetados para la representación de texto.

4. Representaciones propuestas

Las siguientes representaciones toman un texto que es dividido por frases y para cada frase se construye un grafo.

4.1. Representación A

Se basa en establecer las relaciones entre palabras de cada enunciado. Para cada palabra le es asignado un nodo del grafo, este es etiquetado con la misma palabra. Con esta representación se busca mantener la secuencia de los términos en un enunciado. En esta representación las aristas son etiquetadas con la palabra genérica “siguiente” de acuerdo al grado de aparición de cada palabra. La representación no considera una topología base, más bien se adecúa a la estructura de la frase. Con ello se espera encontrar conjuntos de palabras importantes en el texto. En un comparativo con el modelo de n -gramas de palabras no se define un n fijo, más bien es la herramienta de minado la que establece y adapta ese n . Para ilustrar la representación contémplese el siguiente extracto de la

novela de Julio Verne “Viaje al centro de la tierra” “...*me quedé sólo, se me ocurrió la idea de irselo a contar a mi tío, si me quedé sólo ¿Y si mi tío volvía y me llamaba?...*”. La representación basándose en la representación A es la que se observa en la figura 1.

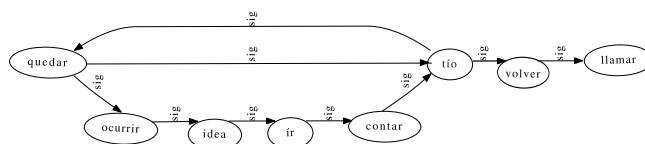


Figura 1. Representación A. Secuencia y orden de las palabras.

4.2. Representación B

La representación B es realizada en base a un etiquetador de dependencias de tipo. Un etiquetador de dependencias de tipo provee una descripción simple de las relaciones en una oración, mostrando las relaciones entre pares de palabras mediante tripletas de la forma *relación(palabra1, palabra2)*. Para realizar este tipo de representación se considera una topología base, el grafo en forma de estrella. La idea principal de esta representación consiste en que los lemas de las palabras son representados como nodos del grafo concatenandoles la clase gramatical, por otra parte las relaciones de dependencia de tipo son las etiquetas de las aristas. Esta representación anula la relación “siguiente” mostrada en la representación A y la substituye por la de dependencias de tipo. En un comparativo con la representación mostrada en [11], en este trabajo usamos el etiquetador de dependencias de tipo para encontrar las relaciones entre pares de palabras tomando considerando una topología base. Esto es con el objetivo de encontrar patrones de escritura con mayor cantidad de estructura y soporte que determinen las características del escritor. Refiérase nuevamente al enunciado anterior, un grafo etiquetado bajo la representación B se observa en la Figura 2.

Una vez que se ha presentado las representaciones se establece la metodología seguida a través de este trabajo en la siguiente sección.

5. Metodología y herramientas utilizadas

Antes de pasar a describir la metodología empleada se describe el conjunto de datos o corpora.

5.1. Conjunto de datos

El conjunto de datos esta conformado por los escritos literarios en español de los autores: C.S. Lewis, Darren Shan, J.K. Rowling, Justin Somper, Julio Verne

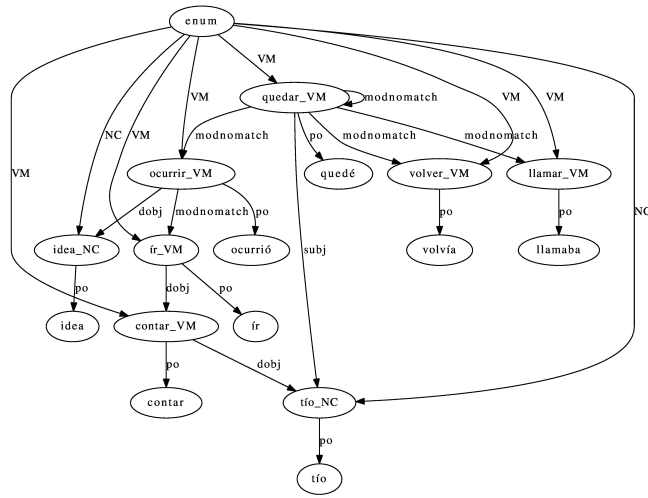


Figura 2. Representación B. Estructura de estrella utilizando un etiquetador de dependencias de tipo.

Tabla 1. Características del corpora considerado.

Característica	C.S. Lewis	Darren Shan	J.K. Rowling	Justin Somper	Julio Verne	Rick Riordan
Num. libros por autor	5	5	5	5	5	5
Tamaño del Vocabulario	37994	29865	88048	33802	72622	44879
Num. de frases	9191	8100	55934	54255	13454	20712
Promedio de palabras por libro	65472	39917	223386	130649	108874	54164

y Rick Riordan, pertenecientes a la literatura de aventura. Un panorama general de las características del corpus se muestra en la tabla 1.

Ahora se describe la metodología empleada que va desde los documentos originales hasta el planteamiento de los modelos de clasificación.

5.2. Metodología

El conjunto de datos es dividido por autor para su tratamiento. La serie de pasos considerado es el siguiente.

1. Cada corpus por autor es separado por enunciados. Entendiendo el concepto de enunciado como una serie de palabras que son separadas por un signo terminal, en este caso el punto.
2. Cada oración es limpiada de signos de puntuación adicionales.
3. Para cada oración se elimina el conjunto de palabras cerradas.
4. Cada oración es pasada por un etiquetador de partes de la oración para obtener el lema de cada palabra (Nota: para el caso de la Representación B se usa un etiquetador de dependencias de tipo.)

5. Se construye un grafo por oración.
6. Se agrupan el conjunto de grafos generados por autor.
7. Las características con mayor relevancia son extraídas del conjunto de grafos usando una herramienta de Minado en grafos.
8. Las características extraídas son colocadas como características de un clasificador supervisado usando Weka¹.

Como etiquetador de partes de la oración se usa Freeling² para el caso de la representación A. En el caso de la representación B se usa el mismo sistema (Freeling), sin embargo se utiliza el parser de dependencias (incluido en la herramienta). Para cada conjunto de oraciones se genera un conjunto de grafos basado en las representaciones propuestas. Este conjunto de datos es importante ya que es la entrada para la herramienta de minería en grafos. La salida de la herramienta de minado es la entrada como características para realizar una clasificación supervisada³.

5.3. Herramientas utilizadas

Con el fin de extraer los patrones de escritura se utiliza la herramienta de minería de datos basada en grafos Subdue. Existen diferentes tipos de herramientas para obtener subestructuras comunes en un conjunto de datos (Subdue [3], gSpan [14], etc.), en este caso de grafos. En este trabajo se usa la herramienta Subdue debido a que su código es libre y ha sido probado en diferentes tareas: descubrimiento de patrones en telecomunicaciones [1], detección de anomalías [4], etc.

Subdue Es usada para descubrir conocimiento, encontrar estructuras y patrones relacionales a partir de la representación mediante grafos etiquetados muy útil en tareas donde se trabaja con dominios estructurados. Subdue usa tres métricas: MDL, la métrica “size” que reporta las mejores subestructuras basándose en el tamaño del grafo y la métrica “cover” que se basa en el número de repeticiones de las subestructuras en un conjunto de grafos.

Ahora que se han descrito todos los elementos necesarios realizados para la extracción de características se presentan los resultados obtenidos en términos del uso de diferentes clasificadores y con cada una de las representaciones propuestas.

6. Resultados obtenidos

La presente sección muestra los resultados de aplicar los métodos tradicionales tanto de características estilométricas (se consideraron: promedio de palabras entre oraciones, tamaño del vocabulario, tamaño de las palabras, número

¹ <http://www.cs.waikato.ac.nz/ml/weka/>

² <http://nlp.lsi.upc.edu/freeling/>

³ Una descripción de las categorías gramaticales se muestra en la página, <http://nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html/>

de mayúsculas y minúsculas, número de signos de puntuación) como el modelo de bolsa de palabras ponderado por frecuencia como comparativo con el modelo propuesto de extracción de características basada en grafos. Las palabras obtenidas con la herramienta de minado son tomadas como características para el clasificador. Se usa la herramienta Weka, en la cual se prueban los conjuntos de características en diversos clasificadores supervisados, entre ellos: Random Forest, Máquinas de vectores de soporte (SVM), árboles de decisión C4.5 y el algoritmo de voto que incorpora la opinión de varios clasificadores. Para el caso particular de este último se consideran dos clasificadores: Random Forest y SVM; se eligen estos dos debido a que son los que aportan una mejor precisión en pruebas individuales. Los resultados están expresados por niveles de precisión. Cabe destacar que los resultados en la tabla 2 reflejan un análisis minucioso de la configuración de los clasificadores, es decir, se prueban diferentes configuraciones para cada uno y los resultados son reportados con las mejores. Se lleva a cabo una prueba de validación cruzada de 10 divisiones como conjunto de prueba. El corpus que se trabaja es el mostrado en la sección 5.1 y los resultados son los siguientes:

Tabla 2. Resultados obtenidos.

Clasificador	Bolsa de palabras (1000 atri.)	Características Estilométricas-6	Representación	Representación
			A	B
Random Forest	0.71	0.65	0.78	0.81
SVM	0.75	0.65	0.79	0.80
C4.5	0.68	0.50	0.63	0.67
Multilayer Pe.	0.66	0.55	0.62	0.67
Voto	0.73	0.69	0.79	0.81

En la siguiente subsección se presenta un análisis meticuloso de los resultados obtenidos resaltando por un lado los niveles de precisión obtenidos por los clasificadores y por otra parte efectuando un comparativo entre los modelos de representación de características.

6.1. Análisis de resultados

Los resultados que figuran en la Tabla 2 muestran la confiabilidad de los clasificadores SVM y el algoritmo de árboles de decisión de Random Forest ya que estos obtienen la mayor precisión global. Enfatizando en las representaciones se muestra que el modelo basado en características estilométricas únicamente logra clasificar correctamente al 65 % de los casos cuando se usa una máquina de soporte vectorial y con otros clasificadores un nivel inferior. Los resultados de las características estilométricas se ven fácilmente superados por el modelo basado en bolsa de palabras cuyo máximo de elementos adecuadamente clasificados se observa en el algoritmo de Voto logrando un 69 % global. En lo que respecta a los resultados obtenidos usando la representación A estos muestran una mejora de

un 6 % en comparación con el mejor resultado obtenido por la bolsa de palabras. Se cree que esto es debido a que la representación A ayuda a plasmar la forma de escritura del autor mostrando las conexiones entre conjunto de palabras. Por otra parte en lo que concierne con la representación B los mejores resultados son los obtenidos con SVM y Random Forest, ya que bajo esta representación se están encontrando características que no sólo reflejan la secuencia de escritura del autor si no que incorporan características como las dependencias de tipo asociadas entre palabras incluyendo patrones léxicos y sintácticos. Esta última representación logra obtener hasta un 81 % en una prueba de validación cruzada.

En la siguiente sección se presentan las conclusiones de este trabajo, así como se realiza una serie de propuestas factibles para seguir explorando la riqueza de una representación basada en grafos.

7. Conclusiones y trabajo futuro

Se presentó una metodología que establece una aproximación para la solución del problema de determinación de autoría. En este trabajo se considera el conjunto de datos de seis autores cada uno con cinco documentos (en este caso libros). La idea fundamental consistió en realizar un proceso de descubrimiento de características fundamentado en la representación de textos basada en grafos. Se han establecido dos propuestas, por una parte la representación A conserva características de secuencia entre los enunciados y por otro lado una representación B la cual incorpora elementos léxicos, sintácticos y las relaciones (aristas) se establecen mediante el uso de las dependencias de tipo. Se realiza una clasificación con base en dos modelos tradicionales bolsa de palabras y características estilométricas de escritura, con el objetivo de realizar un comparativo con los modelos propuestos a través de grafos. Los resultados en términos de precisión muestran una mejora notable al hacer uso de las representaciones propuestas.

Sin lugar a dudas las representaciones basada en grafos fueron un elemento determinante en la búsqueda de características. A través de la minería de datos basada en grafos se pueden encontrar elementos que funjen como atributos para la clasificación. Se puede equiparar las representaciones con los modelos basados en n-gramas de palabras. Sin embargo, los grafos son capaces de proveer combinaciones de características que estén formadas no sólo por unigramas, bigramas o trigramas, sino más bien pueden proporcionar combinaciones notables de características. Además se enriquece la representación incorporando relaciones entre las palabras como es el caso de la representación B, a diferencia de los modelos típicos.

Estas representaciones aún pueden ser mejoradas estudiando las distintas topologías existentes. Además se pueden incorporar conceptos como la sinonimia, hiperonimia, aún no consideradas en estas representaciones. Se debe seguir indagando acerca de las distintas formas en que se pueden relacionar los nodos (palabras), ya que en una representación basada en grafos de esta forma se determina el grado de representatividad de escritura de determinado autor. Considérese que para este trabajo utiliza las representaciones basadas en grafos

como un método para extraer información del texto, por tanto se debe explorar la amplia gama de aplicaciones que las representaciones pueden tener, puesto que en este caso se observó únicamente para el problema de atribución de autoría.

Referencias

1. Baritchi, A., Cook, D.J., Holder, L.B.: Discovering structural patterns in telecommunications data (2000)
2. Castillo, E.: Determinación de características en el proceso de detección de autoría. FCC-BUAP (2012)
3. Cook, D.J., Holder, L.B.: Substructure discovery using minimum description length and background knowledge. *J. Artif. Int. Res.* 1(1), 231–255 (Feb 1994)
4. Eberle, W., Holder, L.: Anomaly detection in data represented as graphs. *Intell. Data Anal.* 11(6), 663–689 (2007), <http://portal.acm.org/citation.cfm?id=1368024>
5. Grieve, J.: Quantitative authorship attribution: An evaluation of techniques (2007)
6. Houvardas, J., Stamatatos, E.: N-gram feature selection for authorship identification. In: *AIMSA*. pp. 77–86 (2006)
7. Luyckx, K., Daelemans, W.: Shallow text analysis and machine learning for authorship attribution
8. Pavelec, D., Justino, E., Batista, L.V., Oliveira, L.S.: Author identification using writer-dependent and writer-independent strategies. In: *Proceedings of the 2008 ACM symposium on Applied computing*. pp. 414–418. SAC '08, ACM, New York, NY, USA (2008), <http://doi.acm.org/10.1145/1363686.1363788>
9. Peng, F., Schuurmans, D., Wang, S.: Augmenting naive bayes classifiers with statistical language models (2003)
10. Potthast, M., Eiselt, A., Barrón-Cedeño, A., Stein, B., Rosso, P.: Overview of the 3rd international competition on plagiarism detection. In: *CLEF (Notebook Papers/Labs/Workshop)* (2011)
11. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic dependency-based n-grams as classification features. In: Batyrshin, I., Mendoza-González, M. (eds.) *Advances in Computational Intelligence, Lecture Notes in Computer Science*, vol. 7630, pp. 1–11. Springer Berlin Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-37798-3_1
12. Sowa, J.F.: *Conceptual structures: information processing in mind and machine*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1984)
13. de Vel, O., Anderson, A., Corney, M., Mohay, G.: Mining e-mail content for author identification forensics. *SIGMOD RECORD* 30, 55–64 (2001)
14. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining (2002)

Clasificación de publicaciones en redes sociales semánticas mediante aprendizaje artificial con redes Bayesianas

J. Carlos Conde-Ramírez, Pablo Camarillo-Ramírez y Abraham Sánchez-López

Facultad de Ciencias de la Computación,
Benemérita Universidad Autónoma de Puebla, México
{juanc.conde,pablo.camarillo,asanchez}@cs.buap.mx

Resumen El estudio de las redes sociales actualmente está relacionado con el desarrollo científico de la Web semántica, dada la gran cantidad de información no estructurada presente en las redes sociales. Ésta información debe ser entendida y clasificada para que los administradores de las redes sociales puedan establecer políticas adecuadas que mejoren la experiencia del usuario. En este trabajo proponemos una metodología para clasificar publicaciones realizadas en una red social semántica desarrollada por los autores. Esta metodología de clasificación está basada en el uso de las redes Bayesianas y en una ontología de dominio para realizar el proceso de anotación semántica de manera más precisa y en forma automática.

Palabras clave: Redes Bayesianas, Web semántica, redes sociales, ontología

1. Introducción

Nuestro estudio parte de la necesidad de mejorar el proceso de anotación semántica de nuestra red social *Moveek* descrito en [1]. El proceso de anotación se explicará con mayor detalle en la sección 2, pero en esencia es el proceso mediante el cual una computadora puede “entender” el contenido que tiene almacenado y asignarle un significado a partir de una base de conocimiento, en este caso una ontología. En este trabajo se presenta una forma de mejorar los resultados obtenidos en el proceso de anotación. Por la naturaleza del proceso de anotación descrito en [1], no todas las publicaciones realizadas en nuestra red social se anotan, ya que la anotación está limitada a la presencia de ciertos términos en el texto que está asociado a la publicación. La manera que proponemos una mejora de este proceso de anotación es el uso de las redes Bayesianas, que nos permitirán clasificar las publicaciones realizadas en nuestra red social a partir de la evidencia y la estadística de la frecuencia de ciertos términos. Con ello se habrá mejorado significativamente la clasificación y por consecuencia el respectivo índice de anotación semántica, haciendo uso de la extracción de la información que depende de la ontología.

En la sección 2 se presentan los preliminares teóricos necesarios para comprender el concepto de la Web semántica y las redes Bayesianas. En la sección 3 se presentan los módulos que se desarrollaron para realizar los experimentos que nos permitieron observar el rendimiento de nuestra propuesta. En la sección 4 se presenta la metodología que proponemos para incrementar el índice de publicaciones anotadas, así como la forma en la que se construye la red Bayesiana utilizada para realizar los experimentos de clasificación. En la sección 5 se presentan los experimentos que se realizaron para obtener el nuevo índice de anotación de este novedoso método de clasificación. En la sección 6 se presentan los resultados del rendimiento de nuestra propuesta para clasificar publicaciones y con ello reforzar el proceso de anotación semántica con ontologías. Finalmente, en la sección 7 se documentan las observaciones y conclusiones que se obtuvieron al desarrollar esta investigación.

2. Marco teórico

A continuación se describen los conceptos más importantes que definen la estructura de esta investigación, como lo son: la Web semántica, la anotación semántica, la ontología y las redes Bayesianas.

Web semántica. La idea es agrupar la información de manera útil y comprensible para la computadora. Por lo tanto uno de los objetivos de la Web semántica es afinar la búsqueda en Internet mediante el uso de metadatos. En este trabajo esos metadatos están contenidos en una ontología, de esta forma, nuestra propuesta es clasificar de manera autónoma la mayor cantidad posible de publicaciones.

Anotación semántica. El término anotación se refiere a una nota, una crítica, una explicación o un comentario. Es decir, escribimos una nota sobre un tema o bien lo criticamos, explicamos o comentamos. Una anotación por sí misma no tiene sentido, está siempre asociada al objeto que ha sido anotado, es por esto que las anotaciones se consideran como *metadatos*. Es importante en este caso precisar el significado del recurso documental; este puede corresponder a un documento completo o bien solamente a un fragmento de este.

Ontología. Una ontología es una conceptualización formal de un dominio, la descripción de sus conceptos y sus relaciones [2,3]. Son modelos de dominio con dos características especiales que conducen a la noción de significado o semántica compartidos donde; las ontologías son expresadas en lenguajes formales con una semántica bien definida, y se basan en una comprensión compartida con la comunidad.

En este estudio, se ha propuesto el desarrollo de una ontología que modele el conocimiento científico que se publica en la red social semántica desarrollada para este estudio[4]. En la sección 3 se presenta la ontología OntoScience como la base de conocimiento, utilizada para la red social semántica que presentamos.

Redes Bayesianas. En inglés *Bayesian Networks* (BN) son una poderosa representación del conocimiento y de los mecanismos de razonamiento. Una red Bayesiana es un modelo probabilístico multivariado, es decir, se vale de métodos

estadísticos para determinar la contribución de varios factores obteniendo un evento simple como resultado. Los eventos y relaciones causales son representados matemáticamente mediante probabilidades condicionales que involucran variables aleatorias que son representadas mediante un grafo dirigido el cual indica la influencia de los factores de forma explícita y que permite obtener la distribución de probabilidad conjunta correspondiente.

La identificación de eventos independientes, por definición, facilita el cálculo de ciertas probabilidades y por lo tanto contribuye a llegar más rápido a una conclusión. Las propiedades gráficas de d-separación¹ se corresponden con las propiedades de independencia en el espacio probabilista asociado. Existe una propiedad fundamental que permite limitar los cálculos de probabilidades, demostrada por Verma y Pear en 1988, que afirma que “Si X y Y son d-separados por Z, entonces X y Y son independientes dado Z” [5].

$$\langle X|Z|Y \rangle \Rightarrow P(X|Z, Y) = P(X|Z)$$

En otras palabras, el bloqueo de información descrito en los grafos de causalidades, es válido también en la representación probabilista subyacente. Por lo tanto la traducción de un grafo causal a un espacio probabilista conduce a resultados consistentes con el razonamiento intuitivo inferido del grafo.

En el *aprendizaje artificial* la recolección de datos generalmente involucra la recopilación de casos, ejemplos, o instancias de objetos de diferentes tipos o categorías (clases), de modo que para el paso de aprendizaje artificial un modelo de estos datos es creado de forma que después pueda ser utilizado para identificar grupos o similitudes en los datos (aprendizaje no supervisado) o predecir la clase de nuevos objetos (aprendizaje supervisado). De la misma forma una BN puede ser creada automáticamente (aprendizaje) usando datos estadísticos (ejemplos). Existen dos tipos de aprendizaje [6]: *Aprendizaje de parámetros* que dada la estructura de una red (grafo), encontrar el mejor conjunto de parámetros (probabilidades condicionales) para considerar los datos observados y *Aprendizaje de estructura* que sin ninguna hipótesis sobre la estructura de la red, buscar aquella que represente lo mejor posible los datos observados una vez que ya se han proporcionado los mejores parámetros. Así, dados los valores de un subconjunto de variables (evidencia) una BN puede calcular las probabilidades de otro conjunto de variables (variables de consulta).

3. Trabajo relacionado

En los últimos años la Web semántica se ha convertido en un área de investigación indispensable para la búsqueda y recuperación de información. En particular artículos como [7,8,9,10] proponen herramientas o metodologías para la extracción y conceptualización de datos en redes sociales.

¹ El concepto de la d-separación permite precisar en qué condiciones una información puede tratarse localmente, sin perturbar el conjunto del grafo. Por lo que la mejor interpretación es el *bloqueo*.

En [11] Q. Rajput y S. Haider proponen el *framework* de notación semántica *BNOSA* para la extracción de información relevante a partir de datos sin estructura, sin una gramática y prácticamente sin coherencia. El conjunto de corpus utilizados corresponden a paginas Web de compra-venta. BNOSA consta principalmente de dos fases para llevar a cabo su propósito. En la primera fase utiliza una ontología previamente definida para extraer los datos y conceptualizar el dominio del problema con la ayuda de *context words* y tipos de datos ya definidos. La segunda fase utiliza los valores *bien definidos* de los atributos producidos en la primera fase como evidencia dura para poder resolver conflictos; como corregir valores duplicados o incluso predecir valores perdidos. Para este fin la segunda fase hace uso una red Bayesiana correspondiente a la ontología definida. Así como esta, existen algunas otras referencias, no menos importantes, sobre métodos de aprendizaje automático en redes sociales como en [12,13,14].

Por otra parte, en investigaciones anteriores se han empleado redes Bayesianas en tareas de *clasificación de documentos*; haciendo un análisis estadístico y utilizando un algoritmo para generar la estructura de la red Bayesiana que resuelva la clasificación de nuevos documentos. Para lo cual debe establecerse cierta relación semántica entre el contenido del documento y su clase. El propósito es observar la clasificación con el modelo de clasificación construido por Naïve Bayes e investigar como el modelo de clasificación con redes Bayesianas mejora la precisión de clasificación generando una estructura más compleja que representa mejor las probabilidades causales. La clasificación es basada en una simple inferencia, es decir, todas las variables excepto la de clase son conocidas. Para comprobar la efectividad de la red se utiliza JavaBayes.

Como ya se mencionó, esta investigación se basa en la red social descrita en [4] la cual utiliza un motor de extracción de información basado en una ontología para realizar el proceso de anotación semántica. El desarrollo de la ontología llamada *OntoScience* permite tener una base de clasificación y por lo tanto de anotación para las publicaciones realizada en nuestra red social. Dicha ontología se realizó tomando en cuenta un dominio científico para que la clasificación de las publicaciones fuera asignada a alguna ciencia y de esta forma, en el proceso de anotación, se tomara esa ciencia como significado de la publicación. En este trabajo utilizamos una muestra de las publicaciones realizadas en esta red social para probar la metodología descrita en la siguiente sección.

4. Metodología propuesta

La metodología propuesta en este trabajo consiste en la creación de una red Bayesiana y su uso para la clasificación de publicaciones realizadas en la red social empleada para este estudio e incrementar el índice de anotación semántica. En la Figura 1 se muestra un esquema general de la metodología para la creación de la red Bayesiana.

Cada una de las siguientes actividades corresponde a un paso en la generación de la red Bayesiana de la Figura 1.

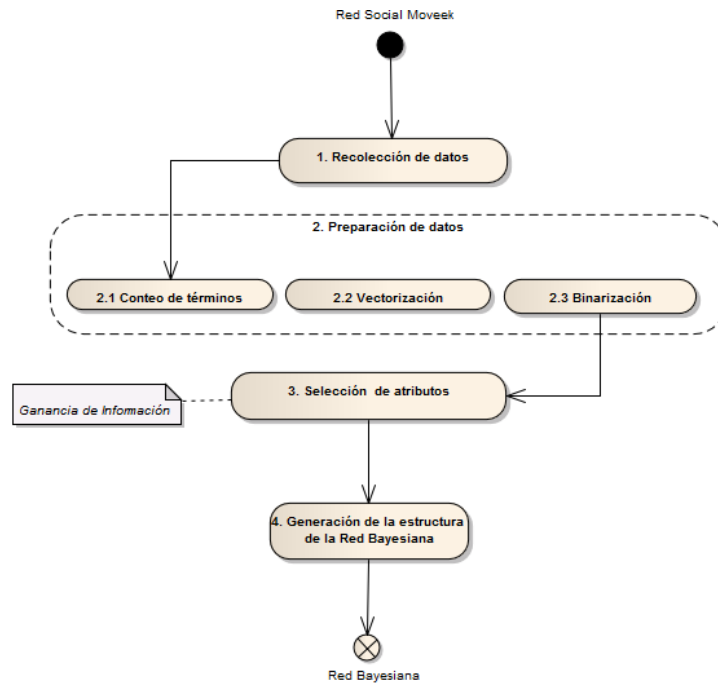


Figura 1. Metodología para la generación de la red Bayesiana

4.1. Recolección de datos

Concentrar todas las publicaciones de la red social, una por renglón, eliminando acentos y signos de puntuación, con la intención de homogeneizar el conjunto de palabras que más adelante serán utilizadas como atributos a seleccionar.

4.2. Preparación de los datos

Una vez concentradas las publicaciones se define el formato del archivo para que este describa explícitamente:

- Nombre del usuario que realizó la publicación (*usuario_publicacion*).
- Cuerpo de la publicación, que corresponde a todas las palabras utilizadas (*contenido_publicacion*).
- Clasificación obtenida por la ontología (*clase_publicacion*).

Con los datos formateados, se continua con el procesamiento de los datos para obtener los atributos o palabras más representativos de nuestro conjunto de datos a través del estudio estadístico pertinente. Esto con la intención de generar la estructura de la red Bayesiana que mejor represente los datos. Este procesamiento de datos consta de los siguientes pasos:

1. Contabilizar términos.

Para lo cual se necesita convertir los datos STRING a NOMINALES y posteriormente VECTORIZAR el resultado. Para vectorización, y por las características de nuestra muestra, fue necesario aplicar una transformación IDF que es una medida para saber si un término es “común” o es “raro” en los documentos. Es obtenida realizando el siguiente cálculo:

$$f_{ij} \log\left(\frac{\text{documentos}}{\text{documentos que contienen el término}}\right)$$

Donde f_{ij} es la frecuencia del término i en el documento j .

2. **Binarizar conjunto de datos.** Con esto los valores de los atributos se convierten en binarios a partir del atributo definido “clase”. Por lo tanto, el nuevo valor de un atributo binarizado será cero sólo cuando su valor original sea exactamente cero. Es decir, aparece o no aparece en la publicación.

4.3. Selección de atributos

Para el proceso de selección de atributos se aplica el método de Ganancia de Información para evaluar un atributo con respecto a su clase. Para predecir la precisión se utiliza el método de Variación Cruzada²; es importante en la vigilancia contra pruebas de hipótesis sugeridas por los datos, especialmente donde más pruebas son difíciles, costosas o imposibles de recolectar [15].

4.4. Generación de la estructura de la red Bayesiana

Como resultado de los procesos antes descritos se obtiene la estructura de la red Bayesiana, la cual nos permitirá clasificar cualquier publicación dada alguna clase de la ontología. A continuación se muestra un breve ejemplo de la estructura en XML que tiene la red Bayesiana empleada para obtener la clasificación de una publicación.

```
1 <BIF VERSION="0.3">
2 <NETWORK >
3 <NAME>FASE2A </NAME>
4 <VARIABLE TYPE="nature">
5 <NAME>su_binarized</NAME>
6 <OUTCOME>v0</OUTCOME> <OUTCOME>v1</OUTCOME>
7 <PROPERTY>position = (100,84)</PROPERTY>
8 </VARIABLE>
9 ...
10 <DEFINITION>
11 <FOR>linguista_binarized</FOR>
12 <GIVEN>clase_publicacion</GIVEN> <GIVEN>con_binarized</GIVEN>
13 <TABLE> 0.83333 0.16666 0.5 0.5 0.9375 0.0625 0.5 0.5</TABLE>
14 </DEFINITION>
15 ...
16 </NETWORK >
```

² Es un método estadístico que evalúa cómo los resultados de un análisis estadístico se generalizan. Utilizado en aplicaciones de predicción, obtiene la precisión un modelo predictivo que se llevará a la práctica.

5. Experimentos realizados

En la fase de aprendizaje *no supervisado*, el análisis estadístico se realizó incrementando el número de atributos seleccionados con el fin de observar el comportamiento de la clasificación con Naïve Bayes y Redes Bayesianas. Inicialmente se utilizaron los algoritmos ya mencionados sin considerar la separación estructural de nuestra ontología; donde las superclases A y B (ciencias fácticas y ciencias formales respectivamente) contienen sus propias subclases. En otras palabras, se consideran sólo a las subclases.

Tabla 1. Estadísticas de clasificación de instancias utilizando validación cruzada.

# atrib.	atributos o términos	% precisión General	% precisión Fase 1	% precisión Fase 2-CA	% precisión Fase 2-CB
1	1	28.57	66.67	81.82	90.0
2	1,2	57.14	66.67	72.73	70.0
3	1,2,3	38.10	71.43	90.91	80.0
4	1,2,3,5	57.14	80.95	90.91	90.0
5	1,2,3,5,7	57.14	80.95	81.82	80.0
6	1,2,3,5,7,9	61.91	76.19	81.82	80.0
7	1,2,3,5,7,9,15	66.67	80.95	81.82	80.0
8	1,2,3,5,7,9,15,20	57.14	80.95	81.82	80.0
9	1,2,3,5,7,9,15,20,25	57.14	80.95	81.82	80.0
10	1,2,3,5,7,9,15,20,25,35	52.38	80.95	72.73	80.0
11	1,2,3,5,7,9,15,20,25,35,45	52.38	80.95	81.82	80.0
12	1,2,3,5,7,9,15,20,25,35,45,55	47.62	76.19	81.82	80.0
13	1,2,3,5,7,9,15,20,25,35,45,55,105	52.38	71.43	72.73	80.0
14	1,2,3,5,7,9,15,20,25,35,45,55,105,155	57.14	71.43	72.73	80.0

En la Tabla 1 se observa que la clasificación en “General”, tiene menor precisión que la obtenida por nuestra propuesta (columnas 4-6). Incluso tomando 7 atributos (precisión máxima de 66.67 %) y aplicando el algoritmo de redes Bayesianas el porcentaje desciende a 61.91 %. Con los resultados en “Fase 1” se observa que con 6 atributos la precisión de clasificación es de 76.19 %. Aplicando el algoritmo de redes Bayesianas a los mismos atributos la precisión sube a 80.95 %. Para los resultados en “Fase 2-CA” se observa que con 6 atributos la precisión de clasificación es de 81.82 %, pero aplicando el algoritmo de redes Bayesianas sube a 90.91 %. Por su parte en “Fase 2-CB” los resultados muestran que con 5 atributos la precisión de clasificación es de 80.0 %, pero aplicando el algoritmo de redes Bayesianas la precisión sube a 90 %.

Nótese que el algoritmo de Redes Bayesianas generalizado no siempre mejora los resultados, por lo que se tomó como referencia el algoritmo de Naïve Bayes. Sin embargo el resultado obtenido con el algoritmo de Redes Bayesianas siempre es aproximado al obtenido con Naïve Bayes.

6. Resultados obtenidos

De acuerdo al comportamiento de los resultados observados en Figura 2, se seleccionaron los atributos de la Tabla 2 para generar los modelos de las redes Bayesianas para la primera y segunda fase.

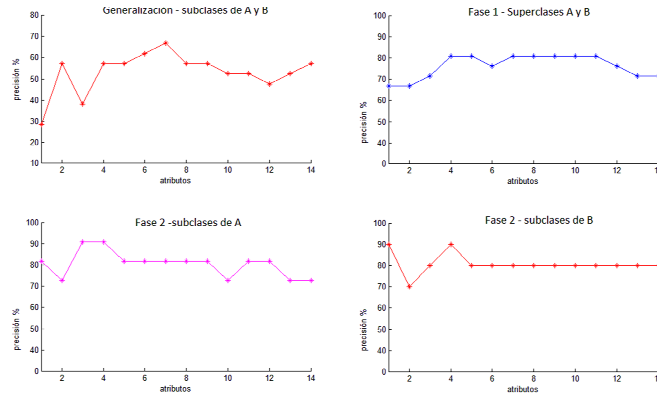


Figura 2. Comportamientos de los resultados obtenidos en las estadísticas.

Una vez generadas las estructuras de las redes Bayesianas que se muestran en Figura 3, lo que resta es asignar evidencia correspondiente para que se calculen las probabilidades relacionadas y así obtener un valor probabilístico de clasificación.

Tabla 2. Atributos seleccionados a partir de la precisión obtenida con el algoritmo de Redes Bayesianas.

Fase de Clasificación	Atributos o Términos	Precisión
Primera, superclases A y B	<i>nuestra, por, hardware, software, programa, sera, son, con</i>	85.7143 %
Segunda, subclases de A	<i>su, con, se, sus, lingüista</i>	90.9091 %
Segunda, subclases de B	<i>sera, virus, problema, mientras, existencia</i>	90 %

La asignación se hace definiendo la presencia o ausencia de cada atributo de la red, representados por los nodos hoja del grafo. En la práctica, cuando un comentario es publicado en *Moveek* y no puede ser clasificado por la ontología, pasa a ser analizado de forma probabilista por las redes Bayesianas. En la primera fase se define a que superclase pertenece. Por ejemplo el comentario:

“programa avanzado de estudios en seguridad informática Versión Julio 2009 El Arte de la Guerra nos enseña que no debemos depender de la posibilidad de que el enemigo no venga sino que debemos estar siempre listos a recibirlo No debemos depender de la posibilidad de que el enemigo no nos ataque sino del hecho de que logramos que nuestra posición sea inatacable”

Las únicas palabras que aparecen son “nuestra” y “programa”, estableciendo esta evidencia se obtiene una probabilidad de 0.9795 de pertenencia a la superclase B (Ciencias Formales), lo cual es correcto. En la segunda fase se observa que ninguna palabra de la publicación aparece sobre la red c) de la Figura 3, por lo tanto la evidencia para cada nodo es de ausencia (valor = falso ó 0).

Sin embargo se obtiene una probabilidad de 0.9385 de pertenecer a la clase B1 (Ciencias de la Computación), lo cual es evidentemente cierto.

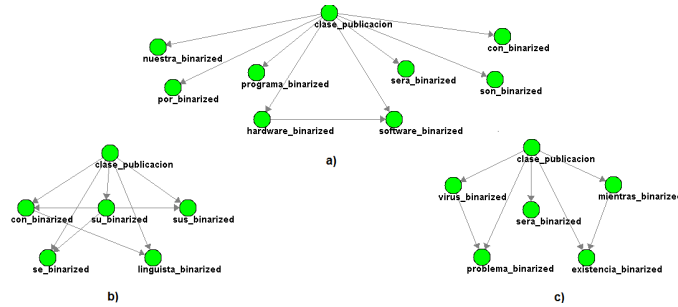


Figura 3. Redes Bayesianas generadas. a)1a. Fase, b)2a. Fase - subclases de A, c)2a. Fase subclases de B

Lo más importante de este enfoque es que tanto la ausencia como la presencia de evidencia contribuyen al cálculo de las probabilidades causales. Esto garantiza la clasificación puesto que siempre se obtiene un valor que define la clasificación confiablemente.

7. Conclusiones y trabajo futuro

Durante el desarrollo de este trabajo de investigación nos pudimos dar cuenta que la tarea de anotación semántica requiere de más de un mecanismo de anotación para incrementar el índice de anotación de nuestra red social. Dado que la metodología propuesta incrementa el número de publicaciones clasificadas en la red social, se cumplió con el objetivo. Sin embargo, los métodos empleados para realizar esta clasificación requieren de una evidencia significativa, es decir, de un número considerable de publicaciones para que los valores inferidos por la red Bayesiana producida sean mucho más confiables.

Por otro lado, éste estudio nos permitió notar que por la naturaleza de la red social, las publicaciones contenidas en dicha red tienen características diversas. Esto hace que el corpus, necesario para el estudio estadístico y la producción de la red Bayesiana, necesite un tratamiento previo para equilibrar el tamaño de las publicaciones agregadas. Por ende, uno de los aspectos que se tomarán en cuenta para mejorar nuestra metodología será establecer un criterio para considerar sólo aquellas publicaciones que permitan obtener un corpus más equilibrado y así mejorar la precisión de clasificación.

Con la metodología propuesta en este trabajo hemos obtenido un proceso de anotación semántica más robusto y exacto, lo cual es un avance significativo para obtener redes sociales completamente semánticas. De esta forma, los administradores de las redes sociales tendrán la oportunidad de conocer mejor a

su audiencia y establecer mecanismos para mejorar la experiencia diaria de los usuarios, entre otras aplicaciones que conlleva lograr construir una red social semántica.

Referencias

1. Pablo, C.R., Abraham, S.L., David, N.R.: Towards a semantic social network. In: IEEE CONIELECOMP 2013. (2013) 74–77
2. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.* **43**(5-6) (December 1995) 907–928
3. Borst, W., Akkermans, J., Top, J.: Engineering ontologies. *International Journal of Human-Computer Studies* (1997) 365–406
4. Pablo, C.R., Abraham, S.L., David, N.R.: Moveek: A semantic social network. In: WILE 2012 (Fifth Workshop on Intelligent Learning Environments). (2012)
5. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1988)
6. Neapolitan, R.: Learning Bayesian networks. Prentice Hall Series in Artificial Intelligence. Pearson Prentice Hall (2004)
7. Mika, P.: Flink: Semantic web technology for the extraction and analysis of social networks. *Web Semant.* **3**(2-3) (October 2005) 211–223
8. Matsuo, Y., Mori, J., Hamasaki, M., Nishimura, T., Takeda, H., Hasida, K., Ishizuka, M.: Polyphonet: An advanced social network extraction system from the web. *Web Semant.* **5**(4) (December 2007) 262–278
9. Garcia-Castro, A., Labarga, A., Garcia, L., Giraldo, O., Montaña, C., Bateman, J.A.: Invited paper: Semantic web and social web heading towards living documents in the life sciences. *Web Semant.* **8**(2-3) (July 2010) 155–162
10. Carminati, B., Ferrari, E., Heatherly, R., Kantarcioglu, M., Thuraisingham, B.: Semantic web-based social network access control. *Computers & Security* **30**(2-3) (March 2011) 108–115
11. Rajput, Q., Haider, S.: BNOSA: A bayesian network and ontology based semantic annotation framework. *J. Web Sem.* **9**(2) (2011) 99–112
12. Andrea, E., Fabrizio, S.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06). (2006) 417–422
13. Grigori, S., Sabino, M.J., Francisco, V.J., Alexander, G., Noé, C.S., Francisco, V., Ismael, D.R., Sergio, S.G., Alejandro, T., Juan, G.: Empirical study of machine learning based approach for opinion mining in tweets. In Batyrshin, I., González Mendoza, M., eds.: *Advances in Artificial Intelligence. Volume 7629 of Lecture Notes in Computer Science.* Springer Berlin Heidelberg (2013)
14. Liu, B.: Sentiment analysis and subjectivity. In Indurkha, N., Damerau, F., eds.: *Handbook of Natural Language Processing, Second Edition.* (2010)
15. Walpole, R.: Probabilidad y estadística para ingeniería y ciencias. Pearson Educación (2007)

Clasificación de acosadores en línea utilizando *Bootstrapping*

Yuridiana Alemán, Darnes Vilariño y David Pinto

Facultad de Ciencias de la Computación,
Benemérita Universidad Autónoma de Puebla, México
{candy.aleman,darnes,dpinto}@cs.buap.com.mx

Resumen En este artículo se presenta una propuesta para el etiquetado automático de un corpus utilizando la técnica de *Bootstrapping*. El corpus utilizado contiene los diálogos de depredadores sexuales en línea, los cuales se catalogan en dos tipos: Los que quieren material pornográfico y aquellos que buscan un encuentro con su víctima. Se utiliza el algoritmo Naive Bayes Multinomial con trigramas de palabras, etiquetando automáticamente un conjunto de entrenamiento y evaluando con una validación cruzada de 10 pliegues. Posteriormente se utiliza el modelo obtenido para clasificar un conjunto de evaluación.

Palabras clave: Bootstrapping, ofensores sexuales, chats, clasificación, Naive Bayes.

1. Introducción

Las conversaciones en línea o “chats” se han popularizado desde la masificación del internet. Los servicios de mensajería instantánea como *Facebook*, *Yahoo* y *Skype* forman parte de la vida diaria de la mayoría de las personas. Entre sus múltiples ventajas se encuentran el costo, rapidez, la sencillez de su manejo y sobre todo, el hecho de que se trata de una tarea que se realiza en privado. Sin embargo, estas ventajas pueden ser contraproducentes si no se tiene el debido cuidado al utilizarlos. Gran parte de los usuarios de este tipo de servicios son jóvenes e incluso niños, por lo que este medio de comunicación puede ser utilizado por personas para cometer delitos, especialmente depredadores sexuales que buscan a sus posibles víctimas mediante redes sociales, utilizando la información personal disponible en la red.

En este artículo se realiza un análisis de un conjunto de diálogos que se encuentran en el idioma inglés obtenidos de conversaciones donde participan ofensores sexuales. Se utiliza la técnica de *Bootstrapping* para clasificar a los ofensores sexuales en dos categorías, según el objetivo que persigan de sus víctimas. Finalmente, se realiza un análisis de los resultados obtenidos.

La estructura del artículo es la siguiente. En la sección 2 se da una descripción de algunas investigaciones alusivas al tema, posteriormente en la sección 3 se explica la técnica de *Bootstrapping* con sus variantes. La sección 4 presenta la metodología aplicada para la investigación, mientras que la sección 5 muestra

los resultados obtenidos. Finalmente, en la sección 6 se explican las conclusiones obtenidas y el trabajo futuro.

2. Trabajo relacionado

La mayoría de las investigaciones respecto al tema toman como punto de partida los experimentos realizados en [1], donde a partir de técnicas de clasificación automática de textos se identifican los diálogos de la víctima y el del depredador utilizando *SVM* y *k-NN*. En los experimentos realizados en [2] se intenta reconocer el *grooming attack*, el cual se define como un “Proceso de comunicación por el cual un autor aplica estrategias de búsqueda de afinidad, mientras que simultáneamente adquiere información sobre sus víctimas con el fin de desarrollar las relaciones que resulten en cumplimiento de su necesidad”. Se utilizaron técnicas de clasificación de documentos para la creación de patrones y detectar en que fase se encuentra la conversación (ganarse a la víctima, cultivo de una relación amorosa ó petición de favor sexual o abuso). Una propuesta diferente para analizar conversaciones es la de [3], donde se identifica cuando ocurre explotación infantil en una conversación. En este trabajo se realiza una comparación entre el uso de características basadas en términos y las extraídas utilizando la herramienta *Linguistic Inquiry and Word Count (LIWC)*[4] para determinar el tipo de conversación que es (explotación infantil, fantasías sexuales entre adultos o chat general sin contenido sexual).

A partir del año 2012 se incluyó la subtarea “*Sexual Predator Identification*” dentro de la conferencia del *Lab Uncovering Plagiarism, Authorship and Social Misuse (PAN)* la cual consiste en determinar a partir de registros de chat, cual es la persona que trata de convencer a otros participantes para proporcionar algún favor sexual e identificar las líneas de las conversaciones de depredadores que son las mas distintivas del comportamiento del mismo. La precisión mas alta fue alcanzada por [5], donde se propone una metodología en dos etapas, una para clasificar las conversaciones donde interviene al menos un depredador sexual y la otra basado en los diálogos por persona para distinguir a los depredadores de las víctimas o pseudo víctimas.

Es importante mencionar que los trabajos relacionados, detectan conversaciones, parte de una conversación, tipo o si el participante es depredador o víctima, pero todas utilizan técnicas de clasificación supervisada y en ninguna se detecta el tipo de depredador, que es el objeto principal del presente artículo de investigación, ya que hasta el momento no existe un corpus etiquetado que funcione como entrenamiento.

3. *Bootstrapping*

En las investigaciones que se caracterizan por la falta de datos, o por datos no etiquetados, las técnicas mas utilizadas son las de *bootstrapping*[6,7,8], las cuales tratan de obtener una gran cantidad de información partiendo de una pequeña “semilla”. Esto es, etiquetar automáticamente un gran número de instancias en

un corpus partiendo de un subconjunto muy reducido de instancias clasificadas manualmente.

Existen muchas técnicas de bootstrapping, las cuales difieren en la manera en que se van agregando instancias al subconjunto etiquetado o las técnicas de selección en caso de utilizarse alguna, sin embargo, todas van de acuerdo al objetivo de la técnica: “*La elevación de un pequeño esfuerzo inicial hacia algo mas grande y más significativo*”.

Algunas de las variantes a esta técnica reportadas en la literatura son:

- ***Self-training***: Esta técnica se utiliza en [6], donde un corpus es utilizado para crear un modelo que se aplica a un conjunto nuevo de frases que tras ser etiquetadas, pasan a formar parte del corpus original, para volver a generar un nuevo modelo y avanzar iterativamente.
- ***Collaborative-training***: Se emplea un mismo corpus para obtener diferentes modelos empleando distintas técnicas de aprendizaje. Posteriormente se introduce una fase de selección entre las diferentes opiniones que surgen de aplicar estos modelos al conjunto de frases nuevas y las etiquetas seleccionadas sirven para aumentar el corpus original y proseguir con la siguiente iteración.
- ***Co-training***: Dos corpus inicialmente iguales sirven para crear dos modelos de diferentes características y los resultados de aplicar estos modelos a un conjunto de frases nuevas se “cruzan”, es decir, las frases etiquetadas por el primer modelo sirven para aumentar el corpus que sirvió para crear el segundo modelo y viceversa. De esta forma un modelo no se alimenta únicamente de su percepción del corpus sino que recibe información de otro modelo que imprime otro punto de vista diferente a la resolución del mismo problema.

En este artículo se utiliza la técnica de *Self-training* para el etiquetamiento del corpus, ya que de las reportadas en la literatura, es mas sencilla al utilizar sólo un modelo para el etiquetamiento de las instancias. Esta técnica se implementó utilizando la herramienta *WEKA*[9].

4. Metodología

Para la realización de los experimentos, se extrajeron diálogos de depredadores sexuales de dos conjuntos de conversaciones distintos:

1. *Perverted Justice*: Es un sitio web que contiene conversaciones entre depredadores sexuales convictos y voluntarios que se hacen pasar por menores de edad.
2. *Training y Test* de la competencia PAN 2012 (<http://pan.webis.de/>): Como se describe en [10], es un conjunto de conversaciones obtenidas de varios repositorios, como son los sitios de <http://omegle.inportb.com/>, <http://www.irclog.org/>, <http://krijnhoetmer.nl/irc-logs/>, además,

de algunas extraídas de <http://www.perverted-justice.com/>. El corpus resultante contiene diferentes tipos de conversaciones de depredadores sexuales, por ejemplo:

- Depredador/Víctima.
- Depredador/Pseudo-Víctima (Voluntario).
- Depredador/Pseudo-Víctima (Policía).

Además, se incluyen conversaciones de otros temas como foros de ayuda, y conversaciones *Adulto/Adulto*, en donde se maneja lenguaje de índole sexual, pero consensuado.

La metodología aplicada para el etiquetamiento automático de los diálogos se muestra en la figura 1.

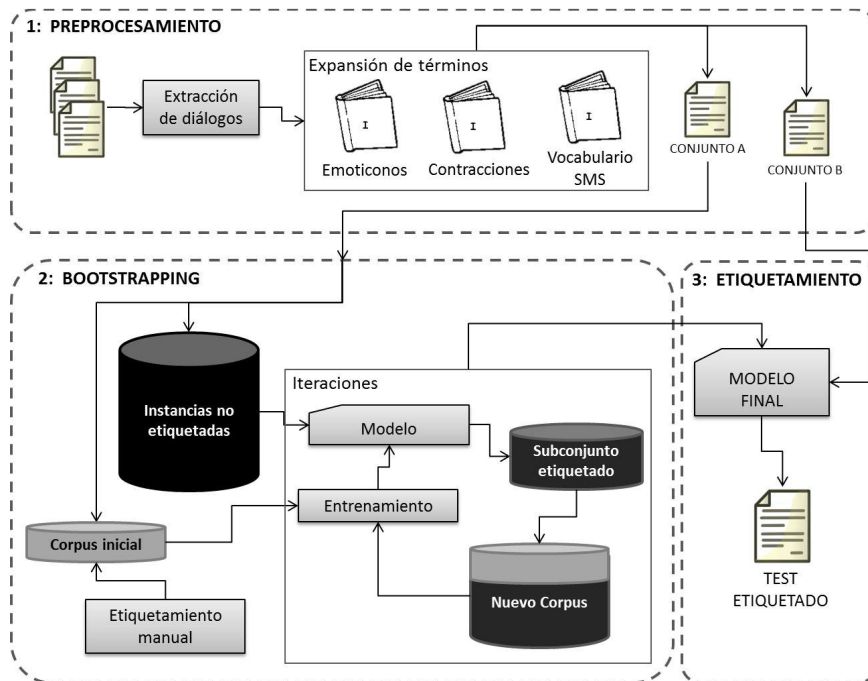


Figura 1. Metodología para el etiquetamiento automático del corpus

Como se puede observar, la investigación se divide en tres fases principales, las cuales se explican a continuación.

4.1. Preprocesamiento

De los conjuntos de conversaciones recopilados, se obtienen solamente los diálogos de los usuarios catalogados como depredadores sexuales, eliminando

el resto de las conversaciones y los diálogos de las víctimas. Finalmente, los diálogos recuperados se unen por usuario (sin importar si son de conversaciones distintas), obteniéndose un conjunto A de 700 diálogos para realizar la clasificación automática, y un conjunto B de 20 diálogos para evaluar el modelo final obtenido.

Gran parte de las diálogos contienen un exceso de términos no reconocidos por un diccionario, además de que abundan los emoticones, cadenas de símbolos raros, que pueden ser URLs, imágenes, entre otros. Dadas estas observaciones, se construyeron 3 recursos léxicos (diccionarios) para ayudar a enriquecer los textos. Estos recursos son:

1. **Emoticones:** Se obtuvo una lista con los emoticones mas comunes (también llamados “smileys”), esta lista fue enriquecida con los emoticones predefinidos de *Windows Messenger*, *Facebook* y *Gmail*. Con esto se lograron recopilar 344 elementos.
2. **Contracciones:** Esta lista contiene alrededor de 65 contracciones mas usadas en los Estados Unidos.
3. **Vocabulario SMS:** Es una lista obtenida de [11], la cual contiene 820 abreviaciones o simplificaciones mas usadas, especialmente por los jóvenes. Estos términos se han ido popularizando en SMS y chats, donde el tiempo de respuesta es importante y generalmente no se toman en cuenta reglas ortográficas y gramaticales.

Tanto en el conjunto A como en el B , todas las ocurrencias de alguno de estos recursos léxicos son sustituidas por su correspondiente significado.

4.2. Aplicación de la técnica

Para utilizar la técnica *Self train* para el etiquetamiento automático, se etiqueta un subconjunto de A manualmente para obtener $A_{etiquetado}$. El etiquetamiento fue realizado por una sola persona, y consistió en leer detenidamente toda la conversación (especialmente los diálogos del depredador) y en base a las peticiones o favores que pedía el depredador a la víctima se asignó una de las siguientes categorías:

- **cat1:** Depredadores que se contentan con obtener y comerciar imágenes de pornografía infantil.
- **cat2:** Depredadores que buscan un encuentro cara a cara con los niños.

De esta forma, el conjunto $A_{etiquetado}$ se conforma de 10 diálogos de depredadores pertenecientes a la categoría 1 y 10 pertenecientes a la categoría 2, estos elementos se eliminan del conjunto A para evitar su doble clasificación. Posteriormente, los 680 diálogos de A se etiquetan con las siguientes reglas:

- Para todas las clasificaciones se utilizan trigramas de palabras, utilizando *LovinsStemmer*.

- Después de hacer experimentos preliminares, se determinó utilizar el clasificador *Naive Bayes Multinomial* para todas las iteraciones.
- Se realizan un total de 6 iteraciones, en cada una se crea un modelo utilizando $A_{etiquetado}$ para clasificar una cantidad de diálogos extraídos aleatoriamente de A .
- Se evalúa la probabilidad de clasificación proporcionada por *WEKA*, si esta es mayor o igual a 90 %, la instancia se elimina de A y se agrega a $A_{etiquetado}$ con su categoría asignada, si esto no se cumple, la instancia no se agrega y continúa en el conjunto A . Al final, se realiza una última clasificación con todas las instancias que no alcanzaron esta probabilidad y se vuelven a clasificar para ser agregadas a $A_{etiquetado}$.
- Las instancias de A extraídas en cada iteración son 30, 50, 100, 200 y 300 respectivamente.

Como se mencionó anteriormente, no se cuenta con un corpus etiquetado, por lo que no hay forma de obtener métricas para la evaluación de las clasificaciones realizadas, sin embargo se realizan evaluaciones en cada iteración con $A_{etiquetado}$ utilizando validación cruzada de 10 pliegues y obteniendo para cada categoría las siguientes métricas:

- *Razón de Falsos Positivos (FPR)*: También denominado ratio o *fall-out*. Se calcula como $FPR = \frac{FP}{FP+VN}$.
- *Razón de Verdaderos Positivos (VPR)*: También denominado recuerdo en recuperación de información, representa la fracción de datos recuperados que son positivos. Se calcula como $VPR = \frac{VP}{VP+FN}$.
- *Exactitud (ACC)*: Representa la fracción de datos evaluados correctamente sobre el total. Se calcula como $ACC = \frac{VP+VN}{Total\ de\ instancias}$.

Donde VP, FN, VN y FP son los datos de la matriz de confusión retornada por *WEKA* (instancias verdaderas positivas, falsas negativas, verdaderas negativas y falsas positivas respectivamente). Con las métricas obtenidas se construye una curva ROC (*Receiver Operating Characteristic*) para la clasificación.

4.3. Clasificación del conjunto B

Una vez etiquetadas todas las instancias de A han sido agregadas a $A_{etiquetado}$, se genera un modelo utilizando nuevamente trigramas de términos y *Naive Bayes Multinomial* para etiquetar los 20 diálogos del conjunto B .

5. Resultados

En la tabla 1 se muestra como se fue etiquetando el conjunto A , tomando como *training* el conjunto $A_{etiquetado}$. En la iteración 1, $A_{etiquetado}$ cuenta con 20 instancias, 10 de cada categoría. Se toman 30 instancias de A como *test*, las cuales se clasifican con el modelo creado por $A_{etiquetado}$. De las 30 instancias clasificadas, 15 tienen una probabilidad de predicción menor al 90 %, por lo que

no se agregan a $A_{etiquetado}$. De las 15 restantes, 9 se agregan a la categoría 1 y 6 a la categoría 2, al término de esta iteración, $A_{etiquetado}$ cuenta con 19 instancias de la categoría 1 y 16 de la categoría 2, las cuales se utilizan como *training* en la siguiente iteración.

En la iteración 2, se extraen 50 instancias de A , de las cuales 20 obtienen una probabilidad de predicción menor y 30 son agregadas al conjunto $A_{etiquetado}$ (17 en la categoría 1 y 13 en la categoría 2), por lo que en la iteración 3, se cuenta con un *training* de 65 instancias. Este proceso se repite hasta etiquetar todo el conjunto A .

Tabla 1. Instancias agregadas al conjunto $A_{etiquetado}$ en cada iteración realizada

Iteración	$A_{etiquetado}$		Diálogos de A	Agregadas		No Agregadas
	Categoría1	Categoría2		Categoría1	Categoría2	
1	10	10	30	9	6	15
2	19	16	50	17	13	20
3	36	29	100	24	33	43
4	60	62	200	45	61	94
5	105	123	300	63	93	144
Final	168	216	316	37	279	-

El conjunto $A_{etiquetado}$ quedó conformado por 205 depredadores de la categoría 1 y 495 de la categoría 2. Como se puede observar el tamaño del test crece considerablemente en cada iteración, así como el número de instancias agregadas a la categoría dos en las últimas iteraciones, este último dato en cierta medida es aceptable, ya que en la realidad existen mas depredadores sexuales que buscan un encuentro con la víctima, además de que son mas peligrosos.

En la tabla 2 se muestran los resultados obtenidos al aplicar las distintas métricas de evaluación.

Tabla 2. Resultados obtenidos por clase en cada iteración

Iteración	Cat 1			Cat2		
	FPR	VPR	ACC	FPR	VPR	ACC
1	0.00	0.80	0.90	0.20	1.00	0.90
2	0.25	0.74	0.74	0.26	0.75	0.74
3	0.25	0.82	0.79	0.18	0.75	0.79
4	0.02	0.85	0.92	0.15	0.97	0.92
5	0.06	0.83	0.89	0.17	0.94	0.89
Final	0.04	0.82	0.92	0.18	0.96	0.92

En todas las iteraciones se observa que las dos razones calculadas son muy similares, y la ACC está arriba del 90% en la mayoría de los casos. En las

iteraciones 2 y 3 la ACC desciende hasta 74 y 79% respectivamente, sin embargo en las siguientes iteraciones se incrementa. Además, la razón de falsos positivos es muy baja, especialmente en la categoría 1.

En la figura 2 se muestran las curvas ROC para las dos categorías en la primera y la última iteración. Como se puede observar, en todos los casos el área de la curva está por encima del nivel medio y sobre todo, no existe una diferencia significativa entre las curvas de la iteración inicial con sólo 20 instancias a las curvas de la iteración final, con 700 instancias.

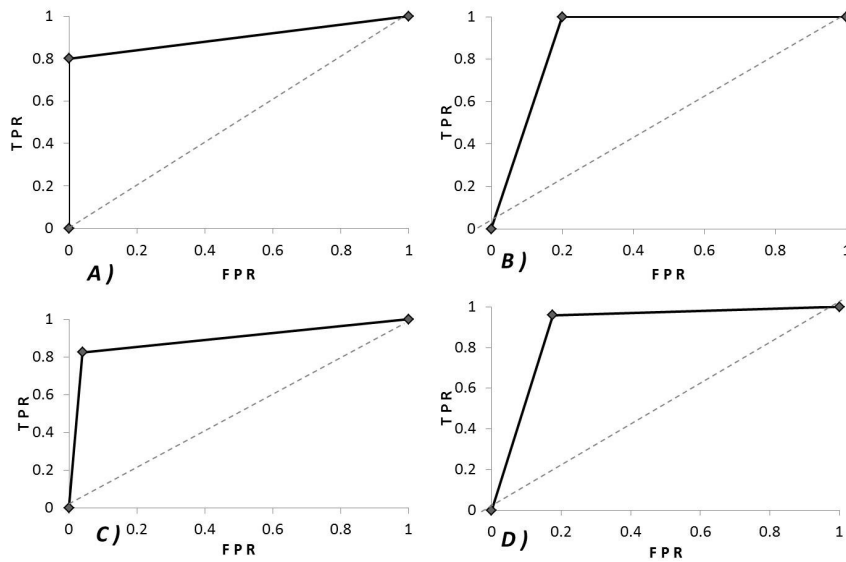


Figura 2. Curvas ROC para el conjunto $A_{etiquetado}$: Categoría 1 (A) y 2 (B) en la iteración 1 y categoría 1 (C) y 2 (D) en la iteración final

Finalmente el etiquetamiento del conjunto B con el modelo creado por los elementos de $A_{etiquetado}$, clasificó 14 diálogos como categoría 1 y 6 como categoría 2. Se eligieron al azar conversaciones de cada categoría para analizarlas manualmente, en las etiquetadas como categoría 1, contienen frases como “i have a pic for you”, “call me”, “send movie”, entre otras. En la categoría 2 se encuentran frases mas relacionadas a sexo explícito. Sin embargo, en algunas conversaciones se crea confusión al momento de realizar la clasificación, ya que el depredador habla de fotos, o hacer video llamadas y posteriormente intenta convencer a la víctima para encontrarse personalmente.

6. Conclusiones y trabajo futuro

En este artículo se presentó una metodología para el etiquetamiento automático de diálogos pertenecientes a depredadores sexuales. Se utiliza la técnica de *Bootstrapping* basado en un pequeño subconjunto etiquetado manualmente. Los resultados obtenidos dan una propuesta de un corpus utilizable para futuros experimentos concernientes a la detección y clasificación de depredadores sexuales en redes sociales.

Como trabajo futuro se busca perfeccionar el etiquetamiento automático utilizando otras versiones de la técnica o el uso de otros conjuntos de características.

Referencias

1. Pendar, N.: Toward spotting the pedophile telling victim from predator in text chats. In: Proceedings of the International Conference on Semantic Computing. ICSC '07, Washington, DC, USA, IEEE Computer Society (2007) 235–241
2. Michalopoulos, D., Mavridis, I.: Utilizing document classification for grooming attack recognition. In: Proceedings of the 2011 IEEE Symposium on Computers and Communications. ISCC '11, Washington, DC, USA, IEEE Computer Society (2011) 864–869
3. Miah, M.W.R., Yearwood, J., Kulkarni, S.: Detection of child exploiting chats from a mixed chat dataset as a text classification task. In: Proceedings of the Australasian Language Technology Association Workshop 2011, Canberra, Australia (December 2011) 157–165
4. Pennebaker, J.W., Chung, C.K., Ireland, M., Gonzales, A., Booth, R.J.: The Development and Psychometric Properties of LIWC2007. This article is published by LIWC Inc, Austin, Texas 78703 USA in conjunction with the LIWC2007 software program.
5. Villatoro-Tello, E., Juárez-González, A., Escalante, H.J., y Gómez, M.M., Pineda, L.V.: A two-step approach for effective detection of misbehaving users in chats. In: CLEF (Online Working Notes/Labs/Workshop). (2012)
6. Clark, S., Curran, J.R., Osborne, M.: Bootstrapping pos taggers using unlabelled data. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4. CONLL '03, Stroudsburg, PA, USA, Association for Computational Linguistics (2003) 49–55
7. Mihalcea, R.: Bootstrapping large sense tagged corpora. In: LREC. (2002)
8. Mihalcea, R.: Co-training and self-training for word sense disambiguation. In: Proceedings of CoNLL-2004, Boston, MA, USA (2004) 33–40
9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. **11**(1) (November 2009) 10–18
10. Inches, G., Crestani, F.: Overview of the international sexual predator identification competition at pan-2012. In Forner, P., Karlgren, J., Womser-Hacker, C., eds.: CLEF (Online Working Notes/Labs/Workshop). (2012)
11. Symens, B.: Acronyms Dictionary for Texting Chatting E-mail. Rebecca J Symens

Visión por computadora y procesamiento de imágenes

Segmentación de objetos en movimiento por flujo óptico y color sin información a priori de la escena

Angel Juan Sánchez García y Homero Vladimir Ríos Figueroa

Facultad de Física e Inteligencia Artificial
Universidad Veracruzana, Xalapa, Veracruz, México.
{angesanchez, hrios}@uv.mx

Resumen Una de las tareas más importantes en el procesamiento de imágenes es la segmentación de objetos. En este artículo una metodología propuesta para la segmentación de objetos en movimiento a partir del flujo óptico y el color, sin requerir información a priori de la escena, es presentada. La estimación del flujo óptico es realizada mediante el algoritmo de Lucas y Kanade de manera piramidal. Los vectores del flujo óptico son agrupados por cercanía, dirección y magnitud. La segmentación del objeto es realizada a partir de la cubierta convexa de los puntos confiables del flujo óptico.

Palabras clave: flujo óptico, segmentación, puntos de interés, color, cubierta convexa.

1. Introducción

En la actualidad, existen aplicaciones cuyo funcionamiento dependen de que un observador, auxiliado de una cámara, sea capaz de identificar los objetos que están en su ambiente. Estas aplicaciones se encuentran en áreas como la robótica, la medicina asistida, el control de tráfico vehicular entre otras. La identificación de objetos en una imagen, suele hacerse por 3 características principales: color, forma y textura. Existen casos donde es requerido conocer la ubicación de los objetos de un color en específico, por lo que los píxeles vecinos que comparten un color similar al especificado son agrupados [5, 6, 14, 25]. También los objetos pueden ser identificados en una imagen por una forma predefinida, por ejemplo, encontrar los píxeles que formen un círculo utilizando la transformación de Hough [2, 13]. Por último, regiones suelen ser identificadas si comparten una textura igual o similar, identificada por diversos métodos como en [4, 15]. Sin embargo, si es requerido identificar objetos que se estén moviendo, es necesario tener como mínimo dos imágenes. Lo anterior es debido a que se debe saber si hubo algún cambio en la intensidad de los píxeles entre las dos imágenes, que nos permita identificar un movimiento. Por lo que, aunque existan otros objetos en un video (o secuencia de imágenes), si estos están estáticos, no deberían ser relevantes. Para identificar los objetos que se movieron de una escena a otra, se suele recurrir al flujo óptico. Horn y Schunck en [10] definieron al flujo óptico como la relación de los cambios espacio-

temporal en la intensidad de los píxeles entre dos imágenes, es decir, encontrar la ubicación de los píxeles que cambiaron de intensidad de una imagen a otra. Para el cálculo del flujo óptico, los métodos generalmente pueden ser clasificados en métodos densos como el de Horn y Shunk [10] y poco densos como el de Lucas y Kanade [16]. El método de Lucas y Kanade se basa solo en información local del píxel, derivada de una pequeña ventana del píxel de interés a diferencia del método de Horn y Shunk que se basa en la información global de la imagen. En ocasiones puede ser requerido no solo la identificación de objetos en movimiento, sino también la creación de un modelo que describa el movimiento, es decir, que la computadora reconozca propiedades del movimiento como su velocidad y dirección.

2. Antecedentes

Como primer antecedente, es mencionada la idea de un sistema que pueda “interpretar lo que ve”. Nagel en [19] propone un CVS (por sus siglas en inglés: Cognitive Vision System), con el cual se espera una descripción en lenguaje natural del desarrollo de una escena del mundo real para tener “una comprensión de lo que está pasando”. Sin embargo debe tenerse conocimiento previo en general, pero también conocimientos específicos de cada dominio del discurso, como posiciones, formas o características de las superficies.

Un problema importante en la segmentación por movimiento, ha sido la identificación de discontinuidades en el movimiento. Este problema ha sido atacado desde diferentes enfoques, como el mostrado por Wang and Adelson en [23], uno de los primeros trabajos donde fue propuesto un enfoque iterativo de crear y poner segmentos para obtener mejores resultados. Posteriormente han habido propuestas para maximizar el enfoque anterior [1, 11, 24]. Desafortunadamente la segmentación suele ser impredecible con estos métodos [26].

Una técnica muy usada es combinar información a priori. Mora y colaboradores proponen en [17] la agrupación de puntos de interés mediante su distribución de manera que satisfagan condiciones geométricas definidas a priori.

La segmentación del movimiento auxiliada por color ha dado buenos resultados [3, 9, 12, 18]. De hecho esta combinación ha sido tan exitosa que es utilizada ampliamente en la segmentación de objetos en visión estereó [22]. Por lo que revisando los trabajos propuestos para segmentar objetos en movimiento, es observado que el color puede ser una herramienta útil para este objetivo.

3. Propuesta

La metodología propuesta es presentada aplicándola a cada par de imágenes en un tiempo t (imagen A) y un tiempo $t+1$ (imagen B). Una vez calculado el flujo óptico por el método de Lucas y Kanade [16] entre las imágenes A y B, los vectores del flujo óptico más representativos de los puntos de interés [21] son obtenidos. Para identificar qué vectores pertenecen a una misma región, partimos de la suposición de que dos vectores pertenecen a una misma región si y solo si cumplen tres condiciones.

La primera condición es que se encuentren cercanos en el espacio 2D, la segunda es que tengan dirección similar, y la tercera que tengan magnitud semejante.

Una vez obtenidas las regiones por flujo óptico (agrupando vectores similares) compuestas por los puntos confiables del flujo óptico, es posible delimitar cada región obteniendo la cubierta convexa de la nube de sus puntos. Puesto que el polígono resultante está definido por sus puntos confiables y agrupa dentro de él algunos puntos del objeto, existen otros puntos, que pertenecen al objeto pero que no están dentro del polígono. Para agregar esos puntos a la región y dar forma similar al objeto, es asumido que esos puntos faltantes tienen color similar a los que están dentro del polígono, por lo que una operación morfológica por color es aplicada partiendo de los puntos interiores del polígono.

4. Metodología

Los pasos a seguir para realizar la segmentación antes mencionada son mostrados en la Figura 1 y descritos a continuación.

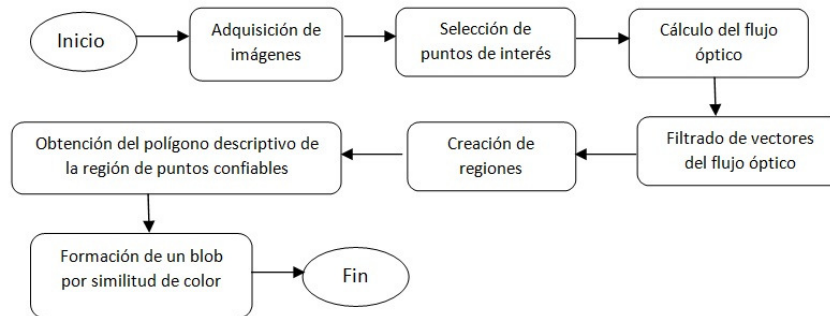


Fig. 1. Diagrama de flujo del algoritmo propuesto.

Para la adquisición de imágenes de los ejemplos mostrados a continuación, fueron adquiridas 10 imágenes por segundo. La resolución de las imágenes adquiridas fue de 2592 x 1944 píxeles.

4.1. Selección de puntos de interés

El cálculo del flujo óptico no puede ser obtenido en todos los píxeles de la imagen debido a que existen píxeles que no nos proporcionan información para ubicarlos en la siguiente imagen (imagen B). Lo anterior es debido a que puede haber píxeles en una superficie de un solo color y sin textura que al ser buscados en la siguiente imagen, varias correspondencias serían obtenidas. Para evitar ese problema, es necesario encontrar píxeles que sean muy diferentes a sus vecinos. A estos píxeles distintivos, Harris [7] los llamó esquinas, cuya característica es tener cambios

considerables de intensidad en direcciones ortogonales. Esta definición está basada en la matriz de derivadas de segundo orden (∂^2x , ∂^2y , ∂x , ∂y) de las intensidades de la imagen, con lo cual es posible generar la conocida matriz Hessiana para cada pixel. Dado que Harris ocupa el método de mínimos cuadrados, el cálculo suele ser costoso. Shi y Tomassi [21] definen a los puntos como “buenos puntos a seguir” basándose en la definición de la matriz de Harris y tomando como criterio para decidir si lo son, el cálculo de los eigenvalores de esa matriz. Al calcular los eigenvalores λ_1 y λ_2 , uno de tres casos posibles es obtenido.

1. Si $\lambda_1 \approx 0$ y $\lambda_2 \approx 0$, entonces el pixel no tiene información relevante.
2. Si $\lambda_1 \approx 0$ y λ_2 tiene un valor positivo grande, entonces se trata de un borde.
3. Si λ_1 y λ_2 tienen valores positivos grandes, entonces en el pixel hay una esquina.

Si el mínimo eigenvalor supera un umbral α , entonces la esquina tiene un valor confiable para ser identificado. En la Figura 2 es mostrada la imagen A y en círculos rojos los puntos de interés encontrados dicha imagen. Puesto que en la mayor parte de la imagen presentada existe una textura casi de sal y pimienta debido al paisaje presentado, pueden ser obtenidos una gran cantidad de puntos de interés.

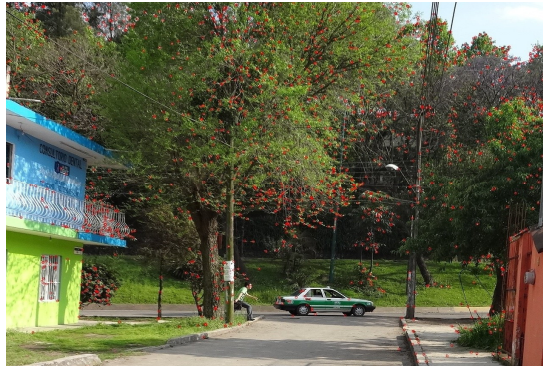


Fig. 2. Puntos de interés en color rojo con $\alpha = 0.01$.

4.2. Cálculo del flujo óptico

Una vez que los puntos confiables son obtenidos, es necesario calcular el flujo óptico, es decir, encontrar la correspondencia de esos puntos en la siguiente imagen. Para el cálculo del flujo óptico fue utilizado el método de Lucas y Kande [16]. Sin embargo, al trabajar con el Método de Lucas y Kanade, se corre el riesgo de que si se usan ventanas pequeñas y el desplazamiento de los objetos es grande, la correspondencia del pixel podría quedar fuera de la ventana. Para evitar este problema, la implementación del algoritmo de manera piramidal fue implementada, donde se empieza a buscar del más alto nivel de la pirámide de la imagen (menos detallada) hasta los niveles más bajos (detalles más finos).

En la Figura 3 son presentados los vectores de flujo óptico. Puede observarse que los vectores creados a partir de los puntos de interés sobre del vehículo, son más grandes en magnitud, puesto que el vehículo presenta un movimiento mayor al de todos los demás objetos que están en la escena.



Fig. 3. Vectores del flujo óptico

4.3. Filtrado de vectores de flujo óptico

Cuando el flujo óptico es calculado, para obtener mayor precisión suele ser procesado a nivel subpixel, lo que suele llevar a la pérdida de información cuando se quiere trabajar a nivel de píxeles posteriormente. Si el desplazamiento ocurrió en menos de un pixel en las dos direcciones, la magnitud del vector será cero. En la segmentación es requerido que por lo menos haya habido un movimiento considerable, por lo que todos los vectores cuya magnitud sea menor a un umbral β_1 , son eliminados. También puede darse el caso de tener vectores muy grandes cuando se calcula el flujo óptico en algunas texturas como las de sal y pimienta [21] (por ejemplo las texturas de pasto y hojas de árboles). Aunque sean buenos puntos de interés, pueden existir varias correspondencias al mismo punto, lo que traería como consecuencia el encontrar desplazamientos muy grandes. Por lo que también deben ser eliminados los vectores cuya magnitud superen un umbral β_2 . Este filtrado es realizado con la finalidad de evitar ruido. En la Figura 4 son mostrados los vectores de flujo óptico después del filtrado, es decir, solo son mostrados los vectores cuya magnitud están en el intervalo (β_1, β_2) , por lo que se concluye que son los vectores que representan el movimiento de los objetos.



Fig. 4. Filtrado de vectores del flujo óptico con $\beta_1 = 1$ y $\beta_2 = 15$.

4.4. Creación de regiones

Para crear cada una de las regiones que definen a los objetos, es necesario agrupar los vectores del flujo óptico con base en 3 criterios: cercanía, dirección y magnitud.

A y B se consideran vectores cercanos si los puntos donde inician los vectores (puntos de interés) no difieren tanto en x como en y de un umbral γ , es decir:

$$\text{abs}(A(x) - B(x)) < \gamma \text{ or } \text{abs}(A(y) - B(y)) < \gamma \quad (1)$$

A y B se consideran vectores de dirección similar si el ángulo entre esos dos vectores no sobrepasa un umbral δ , es decir:

$$\text{ángulo}(A,B) < \delta \quad (2)$$

A y B se consideran vectores similares en cuanto a su magnitud, si la diferencia de sus normas no difiere de un umbral ϵ , es decir:

$$\text{abs}(\|A\| - \|B\|) < \epsilon \quad (3)$$

Con las tres condiciones, son agrupados los vectores de la imagen A en 3 regiones diferenciadas por color mostradas en la Figura 5:

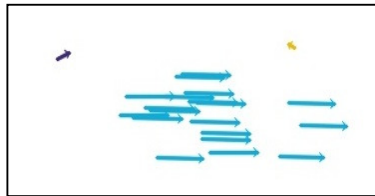


Fig. 5. Imagen que muestra la creación de 3 regiones dadas las condiciones de agrupación con $\gamma = 20$, $\delta = 25$ Y $\epsilon = 10$.

4.5. Obtención del polígono descriptivo del objeto

Una vez agrupado los vectores en regiones, es posible obtener un polígono que represente la cubierta convexa de la nube de puntos de una región. Tomando en cuenta la ecuación de la recta que pasa por dos puntos:

$$y - y_1 = ((y_2 - y_1)/(x_2 - x_1)) (x - x_1) \quad (4)$$

Es posible obtener la siguiente ecuación donde los términos independientes quedan de un solo lado de la igualdad como es mostrado en la ecuación (5).

$$(y_2 - y_1)x - (x_2 - x_1)y = (y_2 - y_1)x_1 - (x_2 - x_1)y_1 \quad (5)$$

La ecuación 5 por lo tanto representa la condición para que un punto (x,y) pertenezca a la recta formada por los puntos (x_1,y_1) y (x_2,y_2) . Por lo tanto podemos representar la condición para saber si dos puntos pertenecen a la cubierta convexa del conjunto de puntos S . Dos puntos $P(x_1,y_1)$ y $Q(x_2,y_2)$ pertenecen al conjunto de la cubierta convexa C si y solo si, todos los puntos $R(x_3,y_3)$ que pertenezca a S (excepto P y Q) al ser evaluados en la ecuación de la recta que pasa por los puntos P y Q , están en un solo lado de la recta (el signo debe tener un solo valor para todo punto R , ya sea positivo o negativo). Es decir, se cumple la ecuación (6) o (7), pero no ambas.

$$P(x_1,y_1) \in C \text{ y } Q(x_2,y_2) \in C \leftrightarrow \forall R(x_3,y_3) \in S \mid (y_2 - y_1)x_3 - (x_2 - x_1)y_3 < (y_2 - y_1)x_1 - (x_2 - x_1)y_1, P \neq Q \neq R \quad (6)$$

$$P(x_1,y_1) \in C \text{ y } Q(x_2,y_2) \in C \leftrightarrow \forall R(x_3,y_3) \in S \mid (y_2 - y_1)x_3 - (x_2 - x_1)y_3 > (y_2 - y_1)x_1 - (x_2 - x_1)y_1, P \neq Q \neq R \quad (7)$$

Esta manera de calcular la cubierta convexa es de orden n^3 , donde n es el número de puntos de la región. Cabe mencionar que para poder identificar un polígono, es necesario que la región tenga por lo menos 3 puntos para cerrar el polígono. En la Figura 6 es mostrado en color rojo el polígono descriptivo del objeto que es delimitado por la cubierta convexa de los puntos de interés que describen el movimiento del automóvil. Dicho polígono está definido por el inicio de los vectores del flujo óptico



Fig. 6. Polígono formado por la cubierta convexa del automóvil en movimiento.

4.6. Formación de un blob por similitud de color

La intención de este paso, es formar un blob (agrupación de pixeles conexos con una forma irregular) que se adapte de mejor manera a la forma del objeto. Partiendo de que los puntos que forman el área del polígono que describe el movimiento del objeto forman también parte del objeto mismo, es posible encontrar los pixeles conexos con color similar mediante un crecimiento de regiones utilizando el algoritmo “Flood fill”. Flood fill (llenado por inundación) [8][20] es una función que a menudo es usada para marcar o aislar partes de una imagen para su procesamiento o análisis. Consiste básicamente en tomar una semilla (el color de un pixel) e ir revisando iterativamente si por lo menos un vecino de ese pixel tiene color similar. Si lo anterior ocurre entonces esos pixeles similares se agregan a la región de la semilla. Este proceso continúa hasta no encontrar al menos un vecino que tenga color similar. Para este trabajo, las semillas para el algoritmo son todos los pixeles que están dentro del polígono descriptivo del objeto. Con este proceso son agrupados todos los pixeles de un objeto, aunque sean de diferente color entre sí, pero similar a cualquiera de los puntos del polígono. Para realizar este proceso, deben ser asignados dos umbrales τ_1 y τ_2 que definan los límites inferior y superior de discrepancia entre el valor cromático del pixel dentro del polígono y de su vecino. En la Figura 7 es presentada la imagen con la aplicación del algoritmo Flood Fill (llenado por inundación) tomando como semillas todos los pixeles del polígono descriptivo del automóvil tomando como parámetros $\tau_1 = 9$ y $\tau_2 = 9$.

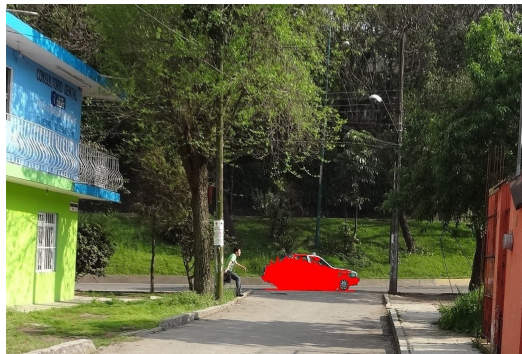


Fig. 7. Resultado de la aplicación de la operación Flood Fill.

5. Resultados

A continuación son mostrados otros resultados del proceso descrito en el artículo. Los parámetros con los cuales fueron obtenidos los mejores resultados (incluyendo los mostrados a continuación) fueron: $\alpha = 0.01$, $\beta_1 = 1$, $\beta_2 = 15$, $\gamma = 20$, $\delta = 25$, $\epsilon = 10$, $\tau_1 = 9$ y $\tau_2 = 9$. En la Figura 8 es presentado otro ejemplo de la metodología presentada.

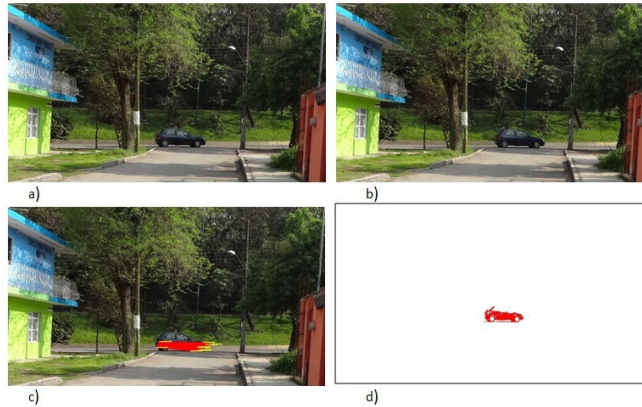


Fig. 8. Las imágenes a y b las capturadas por la cámara, c muestra el polígono con los vectores del flujo óptico y d es la segmentación por color del objeto en movimiento.

6. Conclusiones

La segmentación de objetos por flujo óptico suele ser ambigua debido a diferentes circunstancias como texturas u oclusiones. Con esta propuesta, fue posible segmentar objetos en movimiento en diferentes ámbitos. La importancia de la propuesta radica en que no es requerida información a priori de los objetos móviles que aparecen en la escena, como su forma, color o textura. La identificación de la región que representa a cada objeto es realizada mediante el cálculo de flujo óptico en puntos de interés, lo que disminuye el procesamiento del cálculo. Por último una manera para definir la forma más precisa para cada objeto es realizada mediante semejanza de color de los píxeles que integran el polígono del objeto. Como trabajo futuro podría ser experimentada la creación de un modelo que describa el movimiento de los objetos segmentados en las escenas para inferir sus propiedades.

Referencias

1. Ayer, S., Sawhney, H.: Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. ICCV, pp. 777-784 (1995)
2. Ballard, D. H.: Generalizing the Hough transform to detect arbitrary shapes. In: Pattern Recognition, Vol. 13, Issue 2, pp. 111-122 (1981)
3. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, J., Szeliski, R.: A Database and Evaluation Methodology for Optical Flow. In: International Journal of Computer Vision, Vol. 92, Issue 1, pp. 1-31 (2011).
4. Farrokhina, F., Jain, A.: A multi-Chanel Filtering Approach to Texture Segmentation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1991)
5. Fleye, F.: Color Detection and Segmentation for Road and Traffics Signs. In: IEEE Conference on Cybernetics and Intelligent Systems, Singapore (2004)

6. Fuh, C., Cho, S., Essig, K.: Hierarchical Color Image Region Segmentation for Content-Bases Image Retrieval System. In: IEEE Transactions on Image Processing, Vol. 9, No. 1 (2000)
7. Harris, C., Stephens, M.: A combined corner and edge detector. Proceedings of the 4th Alvey Vision Conference, pp. 147–151 (1988)
8. Heckbert, P.: A Seed Fill Algorithm (Graphics Gems I), New York: Academic Press (1990)
9. Heisele, B., Krebel, U., Ritter, W.: Tracking non-rigid, moving objects based on color cluster flow. In: CVPR, pp. 253 - 257 (1997)
10. Horn, B. K. P., Schunck, B. G.: Determining optical flow. In: Artificial Intelligence 17, pp. 185- 203 (1981)
11. Jepson, A., Black, M.: Mixture models for optical flow computation. In: CVPR, pp. 760-761 (1993)
12. Khan, S., Shah, M.: Object based segmentation of video using color, motion and spatial information. In: CVPR, pp. 746- 751 (2001)
13. Leavers, V. F.: Which Hough Transform?. In: CVGIP: Image Understanding. Vol. 58, pp. 250- 264 (1993)
14. Littman, E., Ritter, H.: Adaptive Color Segmentation - a Comparison of Neural and statistical Methods. In: IEEE Transactions on neural networks, Vol. 8, No. 1 (1997)
15. Lu, C., Chung, P., Chen, C.: Unsupervised texture segmentation via wavelet transform. In: Pattern Recognition, vol. 30, issue 5, pp. 729 - 742 (1997)
16. Lucas, B. D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 1981 DARPA Imaging Understanding Workshop pp. 121-130 (1981)
17. Mora, D., Páez, A., Quiroga, J.: Detección de objetos Móviles en una Escena utilizando Flujo Óptico. In: XIV Simposio de Tratamiento de Señales, Imágenes y Visión Artificial, Colombia (2009)
18. Mukherjee, D., Deng, Y., Mitra, S.: Region based video coder using edge flow segmentation and hierarchical affine region matching. In: SPIE, Visual Communications and Image processing, vol. 3309, pp. 338-49 (1998)
19. Nagel, H., Steps towards a cognitive vision System. IAKS Karlsruhe.
20. Shaw, R.: QuickFill: An efficient flood fill algorithm. <http://www.codeproject.com/gdi/QuickFill.asp>.
21. Shi, J., Tomasi, C.: Good features to track. In: 9th IEEE Conference on Computer Vision and Pattern Recognition (1994)
22. Tao, H., Sawhney, H., Kumar, R.: A global matching framework for stereo computation. In: ICCV, pp. 532-539 (2001)
23. Wang, J., Adelson, E.: Representing moving images with layers. In: IEEE Transactions on Image Processing, vol. 3, no. 5, pp. 625-638 (1994)
24. Weiss, Y., Adelson, E.: A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In: CVPR, pp. 321-326 (1996)
25. Wu, Y., Liu Q., Huang, S.: An Adaptive Self-Organizing Color Segmentation Algorithm with Application to Robust Real-Time Human Hand Localization. In: Asian Conf. on Computer Vision, Taiwan (2000)
26. Zitnick, C. L., Jojic, N., Kang, S. B.: Consistent Segmentation for Optical Flow Estimation. In: Proceeding ICCV proceedings of the tenth IEEE International Conference on Computer Vision, Vol. 2, pp. 1308 - 1315 (2005)

Recuperador de imágenes en base a contenido: una aplicación con esporas

Nayeli Joaquinita Meléndez Acosta, Homero Vladimir Rios Figueroa
y Antonio Marín Hernández

Departamento de Física e Inteligencia Artificial,
Xalapa, Veracruz, México
nayelimelendez@gmail.com, hrios@uv.mx, anmarin@uv.mx

Resumen La identificación taxonómica por género y especie de los hongos micorrízicos arbusculares (HMA) tiene su fundamento en los detalles de cada estructura, como son color, textura, dimensiones y forma. En este documento, se explica el desarrollo de un sistema de recuperación de imágenes basada en contenido para ser utilizado en el área de la microbiología empleando esporas. Para el color se utilizaron los Histogramas, usando el espacio HSV, en textura se aplicaron 3 características de Tamura y en forma 4 Momentos Invariantes Afines. La base de datos contiene 225 imágenes con 60 clases, cada clase tiene de 1 a 6 esporas. 3 experimentos se realizaron uno por característica. En promedio nuestro sistema realiza un diagnóstico correcto del 90 % y una predicción de clasificación del 87 %. Esta propuesta se enfoca a un área donde aun no se cuenta con una herramienta similar según expertos del área, lo que nos llevo a realizar dicho sistema.

Palabras clave: hongos micorrízicos arbusculares (HMA), momentos invariantes afines, características de Tamura.

1. Introducción

Debido al aumento excesivo en la colección de imágenes digitales, son requeridas búsquedas eficientes de imágenes, navegación y herramientas de recuperación por los usuarios de dominios diferentes [1].

Este trabajo se enfocara en la recuperación de imágenes basada en contenido (Content-based Image Retrieval o CBIR), es decir la búsqueda analizará los contenidos reales de la imagen. El término contenido se refiere a colores, formas, texturas o cualquier información que se derive de la imagen.

Los sistemas CBIR se basan en características visuales de las imágenes como el color, el brillo, las texturas y su distribución espacial (que describen el contenido de la imagen) para reconocer patrones similares en las distintas imágenes.

Los hongos micorrízicos arbusculares (HMA) son microorganismos del suelo que son simbioses obligados de las plantas. Su estudio siempre está ligado a la microscopía, sobre todo en el campo de la taxonomía. La taxonomía clásica se basa en el estudio microscópico de las esporas y estructuras de estos hongos,

cuyas dimensiones varían entre las 300 y 20 μm . La identificación taxonómica por género y especie tiene su fundamento en los detalles de cada estructura, como son color, textura, dimensiones, forma y complejidad de la estructura [2]. Las técnicas de identificación taxonómica de los HMA requieren de un conocimiento especializado, sobretodo de la experiencia y entrenamiento en la identificación de estructuras; este proceso suele ser tedioso y requiere tiempo, algunas veces hasta días solo para separar una especie de una muestra de varios individuos. La biología molecular es una técnica que, aunque precisa, es costosa y requiere de infraestructura y un entrenamiento muy especializado del personal.

Los procesos de análisis de imágenes para clasificar objetos han demostrado ser una herramienta útil en el ahorro de tiempo dentro de los procesos de investigación científica. Las ventajas de tener software especializado que nos permita separar o clasificar no solo representan un ahorro de tiempo, sino que también permite facilitar el proceso de enseñanza a las generaciones nuevas, que pretendan estudiar los HMA y especializarse en la rama de taxonomía.

La utilidad de un software que permita clasificar e incluso identificar especies de HMA mediante el análisis de imágenes, abriría área de oportunidad dentro del área de estudio.

2. Fundamentos

Los sistemas CBIR realizan las consultas mediante una imagen de referencia, la cual contiene las características visuales que deseamos buscar, con esta imagen, el sistema realiza el procesamiento necesario para extraer el vector de características el cual no es más que un conjunto de descriptores que forman el índice de la imagen. Este vector es comparado con los vectores de características de cada una de las imágenes que conforman la base de datos. La comparación se realiza mediante una métrica o función de similitud que permite recuperar una lista de imágenes que se aproximen mejor a la consulta realizada. La Figura 1 ilustra el diagrama del proceso de consulta, el cual representa la estructura básica de un sistema CBIR [4].

Los principales componentes que constituyen un sistema CBIR son:

Interfaz de usuario: debe permitir realizar las consultas y visualizar los resultados de las mismas. Las consultas pueden ser realizadas mediante imágenes de referencia dadas al sistema. Esta permite la interacción del usuario para refinar el proceso de búsqueda a partir de la realimentación de valores para obtener el resultado deseado.

Análisis de imágenes: se encarga de realizar el procesamiento de las imágenes, su análisis y extracción de las características mediante técnicas de visión. Aquí se calcula y cuantifica la información relativa al color, a texturas o formas presentes en la imagen.

Comparación y funciones de similitud: este es un aspecto fundamental y se relaciona con la definición de los vectores de características. Las métricas o funciones de similitud permiten comparar el vector de características de la imagen de referencia con los vectores de características de las imágenes almacenados en

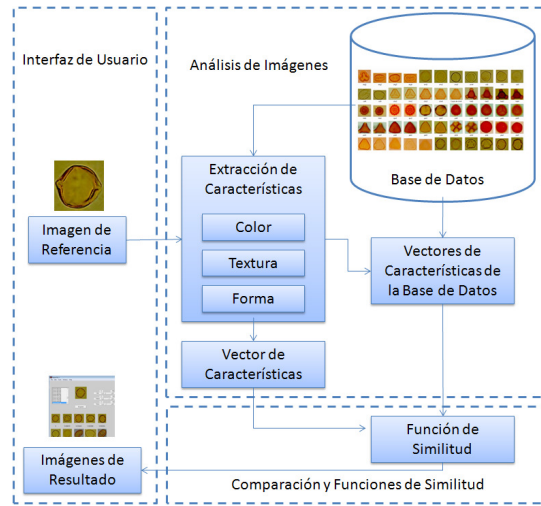


Figura 1. Esquema de una consulta en un sistema CBIR.

la base de datos del sistema, y establecer cuáles son más cercanas o similares a la consulta dada.

3. Arquitectura del sistema

Ahora se describen los métodos utilizados en el proceso de extracción de características primitivas de la imagen, que es el aspecto clave en los sistemas CBIR.

3.1. Color

Esta características es uno de los aspectos más representativos de la imagen. Aunque la mayoría de las imágenes están en el espacio de color RGB, no es éste el más adecuado para la obtención de su caracterización cromática, ya que es el espacio de color más ajeno a la percepción del ser humano.

Histogramas de Color: Una forma de extraer esta característica de color es mediante el uso de histogramas. Los histogramas de color son una forma que representa la distribución del color en imágenes, donde cada bin del histograma representa un color en el espacio de color [5].

Espacio de Color: El espacio de color determina la manera en como representamos los diferentes colores. Existe una gran variedad de espacios de color, en este trabajo se utilizó el espacio HSV (Hue-Saturation-Value) que define los colores en base a tono (hue), cantidad de color presente (saturation) y brillo (value), esta elección se hizo en base a los resultados en [1] donde HSV es mejor que RGB en el contexto de cambios de iluminación, además en este espacio las

diferencias de color son más cercanas a las diferencias halladas en la percepción de los humanos.

Métrica de similitud en Color: Una de tantas métricas utilizadas para medir la similitud entre histogramas es la chi cuadrada [5] la cual es utilizada en este trabajo, esta métrica da un valor pequeño si las imágenes son similares y un valor grande en caso contrario. En dos imágenes idénticas da un valor de 0.

$$d_{chi\ cuadrada}(H_1, H_2) = \sum \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)}. \quad (1)$$

donde, H1 y H2 son los histogramas a comparar.

3.2. Textura

La textura es otra característica importante de los objetos que nos permite medir la regularidad de una imagen. La textura es definida por un conjunto de pixeles, es decir un solo pixel no define algún tipo de textura [5].

Características de Tamura: Las características de Tamura son en total 6, y sus autores encontraron que hay tres de ellas que correlacionan fuertemente con la percepción humana: coarseness (se entiende como rugosidad), contraste y direccionalidad. En su trabajo original en [6] presenta la forma de calcular las 6 características para obtener un valor escalar por cada imagen procesada.

1. *Tamura Coarseness:* El coarseness proporciona información acerca del tamaño de las texturas en la imagen. Cuando las texturas tienen un micro patrón y un macro patrón el procedimiento de Tamura considera el patrón más grande, y la forma de hacerlo es aplicando operadores de distinto tamaño de la siguiente manera:

Sea I una imagen; $I(x, y)$ el valor del píxel en la posición (x, y) .

- a) Para cada píxel (x, y) se calcula el promedio sobre una vecindad definida que debe ser de un tamaño potencia de dos, por ejemplo $2 \times 2, 4 \times 4, \dots, 32 \times 32$:

$$A(x, y) = \frac{1}{2^k} \sum_{i=1}^{2^k} \sum_{j=1}^{2^k} I(x - 2^{k-1} + i, y - 2^{k-1} + j) \quad (2)$$

- b) Para cada punto (x, y) se calcula la diferencia entre las vecindades que no se solapan a lados opuestos del punto en las direcciones horizontal y vertical.

$$D_h^k(x, y) = |A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)| \quad (3)$$

o

$$D_v^k(x, y) = |A_k(x, y + 2^{k-1}) - A_k(x, y - 2^{k-1})| \quad (4)$$

- c) Para cada punto (x, y) seleccionar el valor con la diferencia más grande:

$$S(x, y) = \text{Max}_{d=h,v} \{E_k^d(x, y)\}, k = \{1 \dots 5\} \quad (5)$$

- d) Finalmente tomar el promedio 2^S como una medida de rugosidad para la imagen:

$$F_{crs} = \frac{1}{nm} \sum_{x=1}^n \sum_{y=1}^m 2^{S(x,y)} \quad (6)$$

2. *Contraste de Tamura*: El contraste está influenciado por los niveles de grises, la polarización en los niveles de blanco y negro, definición de los bordes y período de repetición de los patrones. Este se calcula utilizando la media y la varianza de los valores de intensidad en una vecindad dada de la siguiente manera:

$$F_{con} = \frac{\sigma}{\alpha_4^z} \text{ con } \alpha_4^z = \frac{\mu_4}{\sigma^4} \quad (7)$$

en donde

$$\sigma^4 = \frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y (I(x,y) - \mu)^4 \quad (8)$$

$z = 0,25$ es recomendado como el mejor discriminante de texturas

3. *Direccionalidad de Tamura*: La presencia de dirección en la imagen es más interesante, que la orientación. Para calcular la direccionalidad, las derivadas horizontal y vertical son calculadas al convolucionar la imagen I utilizando los siguientes operadores de 3×3 :

$$\Delta_H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ y } \Delta_V = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (9)$$

Luego, para cada posición (x, y) se calcula el ángulo de orientación utilizando:

$$\theta = \frac{\pi}{2} + \tan^{-1} \frac{\Delta_V(x,y)}{\Delta_H(x,y)} \quad (10)$$

Con la definición de las características de Tamura presentadas anteriormente, se obtienen tres medidas de textura por imagen, con los cuales puede construirse el vector de características que contiene información de las texturas, para construir un vector de Tamura.

Métrica de Similitud en Textura: La medida de similitud de textura entre dos imágenes está representada de forma cuadrática entre el vector característica de la imagen de entrada y el vector característica de una imagen de la base de datos (Distancia Euclidiana), la cual es definida como:

$$d = \sqrt{(f_{11} - f_{12})^2 + (f_{12} - f_{22})^2 + \dots + (f_{1n} - f_{1n})^2} \quad (11)$$

donde, f_{ij} esta definida como una característica de Tamura

3.3. Forma

Para el análisis de forma existe el uso de momentos invariantes afines los cuales son invariantes a traslación, rotación, escala y transformadas afines.

Momentos Invariantes Afines: El interés de los momentos generales está en que los contornos pueden modelarse como un tipo especial de funciones acotadas y calcular los momentos de la misma, es por esto que se aplica en el reconocimiento de formas, los momentos invariantes clásicos fueron presentados por [8]. En [9] derivaron un nuevo conjunto de momentos invariantes que son invariantes bajo transformadas afines, este trabajo hace uso de 4. *Momentos Geométricos y Momentos Centrales:* La definición de los momentos geométricos 2-D de orden $(p + q)$ de una funciones de distribución de densidad $f(x, y)$ como la siguiente:

$$m_{pq} = \sum \sum x^p y^q f(x, y) \quad (12)$$

Los momentos que tiene la propiedad de traslación invariante son llamados momentos centrales y son definidos como:

$$\mu_{pq} = \sum \sum (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (13)$$

donde \bar{x} y \bar{y} son las coordenadas del centroide de la función de la imagen $f(x, y)$

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \quad (14)$$

Momentos invariantes afines implementados:

$$Inv_1 = \frac{1}{\mu_{00}^4} (\mu_{20}\mu_{02} - \mu_{11}^2) \quad (15)$$

$$(16)$$

$$Inv_2 = \frac{1}{\mu_{00}^6} (\mu_{40}\mu_{04} - 4\mu_{13}\mu_{31} + 3\mu_{22}^2) \quad (17)$$

$$Inv_3 = \frac{1}{\mu_{00}^7} (\mu_{20}\mu_{21}\mu_{30} - 4\mu_{20}\mu_{12}^2 - \mu_{11}\mu_{03}\mu_{30} + \mu_{11}\mu_{21}\mu_{12} + \mu_{02}\mu_{30}\mu_{12} - \mu_{02}\mu_{21}^2) \quad (18)$$

$$Inv_4 = \frac{1}{\mu_{00}^{10}} (\mu_{30}^2\mu_{02}^2 - 6\mu_{21}\mu_{12}\mu_{30}\mu_{03} - 4\mu_{30}\mu_{12}^3 + 4\mu_{03}\mu_{21}^3 - 3\mu_{21}^2\mu_{12}^2) \quad (19)$$

Métrica de Similitud en Forma: La métrica utilizada para comprar dos vectores de forma es la Distancia Euclidiana. Si la variación se acerca a 0 significa que son similares en caso contrario no se parecen.

3.4. Recuperación de imágenes usando color, textura y forma

El resultado de una recuperación usando más de una características puede ser mucho más eficiente, por ello uso la combinación de tres características color, textura y forma.

La similitud entre la imagen de referencia y una de la base de datos es medida usando tres características: color, textura y forma. Estas características representan diferentes aspectos por lo que es conveniente agregar un peso que diferencie su valor. La distancia entre dos imágenes fue tomada de [10] y es medida como:

$$d = w1 * d1 + w2 * d2 + w3 * d3 \quad , \quad \sum_{i=1}^3 w_i = 1 \quad (20)$$

donde, $w1$ es el peso de la característica de color, $w2$ es el peso de la textura, $w3$ es el peso de la forma y $d1, d2$ y $d3$ son las distancias calculadas de color, textura y forma respectivamente entre las imágenes. La suma de los pesos es un promedio ponderado.

4. Resultados experimentales

Este sistema persigue permitir al usuario clasificar e incluso identificar especies de HMA mediante el análisis de imágenes. Se realizaron los experimentos haciendo uso de la base de datos de imágenes en [11]. La Figura 2 muestra algunas imágenes de la base de datos utilizada.

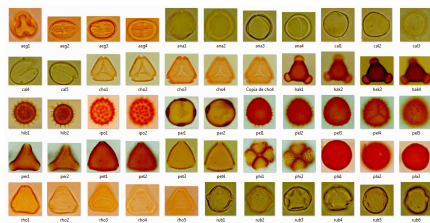


Figura 2. Base de datos de esporas.

Las imágenes utilizadas en la experimentación contienen un solo objeto esto permite que la extracción de características sea más fácil, principalmente en forma. La base de datos tiene 225 imágenes, repartidas en 60 clases, cada clase tiene de 1 a 6 imágenes. El usuario del sistema tiene la elección de elegir el peso de cada característica.

La Figura 3 muestra los resultados de 3 experimentos, un experimento por característica: La Figura 3 (a) ejemplifica el color con un porcentaje de uso más alto que el de textura y forma, los resultados son invariantes a color mostrando así que la característica más importante para esta consulta es el color. En la Figura 3 (b) se puede observar claramente el buen desempeño de los Momentos Invariantes Afines, la consulta se realiza usando forma al 100%, el resto de las características no son tomadas en cuenta, en los resultados se puede ver en la

primera fila que las esporas uno, tres, cuatro y cinco pertenecen a la misma clase y que ni color, ni textura influyen en la recuperación de la espora dos. En el último experimento mostrado en la Figura 3 (c) el porcentaje de uso de textura es el más alto y se puede observar como que los resultados son buenos ya que predomina la textura, incluso el color se preservó.

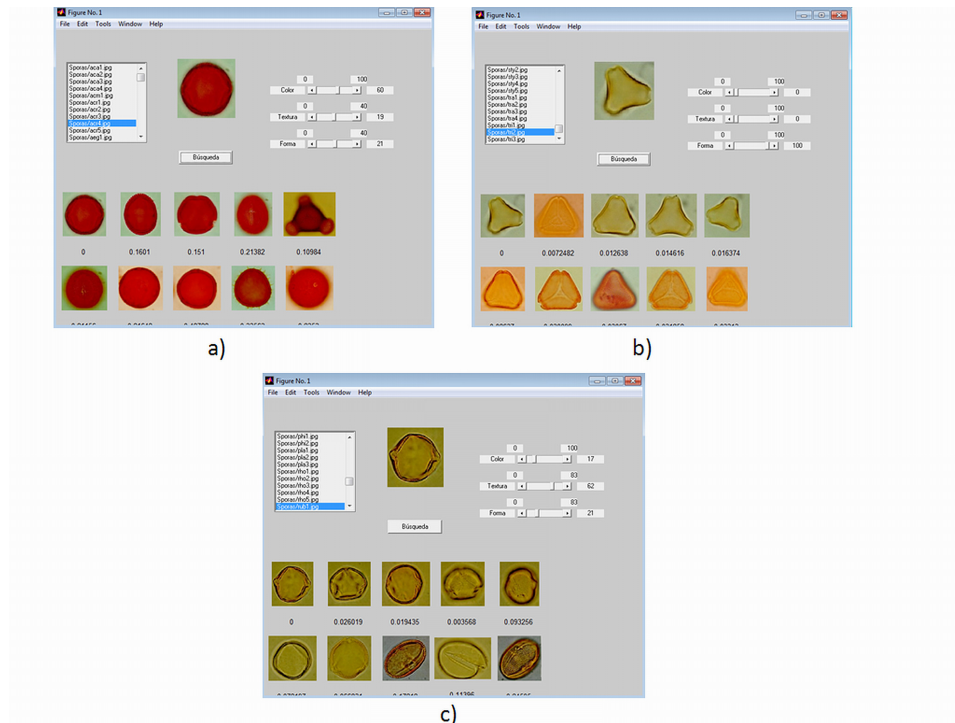


Figura 3. a) Espora=acronychia_oblongifolia, Color=60%, Textura=19% y Forma=21% b) Espora=tristaniopsis_laurina, Color=0%, Textura=0% y Forma=100% c) Espora= rubus_fruticosus, Color=17%, Textura=62% y Forma=21%.

4.1. Evaluación y comparación

Generalmente los sistemas CBIR son evaluados con relación a su rendimiento usando la observación humana. Nuestro sistema de CBIR primero es evaluado en términos de efectividad de recuperación. Para evaluar la efectividad de los sistemas de recuperación, son usadas dos métricas bien conocidas:

$$\text{Precisión} = \frac{\text{Número de imágenes relevantes recuperadas}}{\text{Número total de imágenes recuperadas}} \quad (21)$$

$$\text{Recuperación} = \frac{\text{Número de imágenes relevantes recuperadas}}{\text{Número total de imágenes relevantes en la base de datos}} \quad (22)$$

En la evaluación 5 expertos con conocimiento en esporas fueron jueces, los cuales valoraron los resultados y expusieron si las imágenes resultantes son similares a la imagen de referencia, la decisión de similitud es dominada por la mayoría de votos. Se tomaron en cuenta 3 de varios experimentos realizados, los cuales ya han sido mostrados en la Figura 3. El cuadro 1 muestra los resultados obtenidos en los experimentos, en promedio nuestro sistema realiza un diagnóstico correcto (precisión) del 90 % y una predicción de clasificación (recuperación) del 87 %, proporcionando buenos resultados.

Tabla 1. Evaluación de Efectividad

Experimento	Espora	Precisión	Recuperación
1	acronychia_oblongifolia	0.9	0.81
2	tristaniopsis_laurina	1	0.94
3	rubus_fruticosus	0.8	0.88
Promedio		0.9	0.87

En el cuadro 2 se observa una comparativa entre sistemas tomando en cuenta las características que usan: Color, Textura, Forma y Palabras Clave, y que tipo de sistema es: específico o general. CBIR-Esporas usa tres características y es específico aunque puede ser general si la base de datos cumpla con las especificaciones requeridas, estas características hablan bien de nuestro sistema.

Tabla 2. Comparativa de sistemas.

Sistema	Color	Textura	Forma	Palabras clave	General	Específico
QBIC	*	*	*	*	*	
PicToSeek	*		*		*	
Photobook	*	*	*	*	*	
CIRES	*	*				*
NETRA	*	*	*		*	
Simplicity	*	*	*			*
Cortina	*	*		*	*	
SIMBA		*			*	
Alta Vista	*	*	*	*	*	
CBIR-Esporas	*	*	*		*	*

5. Conclusiones y trabajo futuro

El método propuesto usa color, textura y forma para crear un sistema de recuperación de imágenes. La recuperación de imágenes de esporas puede usarse para la educación e investigación ecológica. Usando las tres características el sistema proporciona mejores resultados. Además esta propuesta se enfoca a un área donde aun no se cuenta con una herramienta similar según expertos del área, lo que nos llevo a realizar una adaptación innovadora del conocimiento. Los resultados obtenidos en los experimentos muestran que en promedio el sistema realiza un diagnostico correcto del 90% y una predicción de clasificación del 87%, proporcionando buenos resultados. En un futuro, probaremos con una base de datos de imágenes de HMA que será proporcionada por el Laboratorio de Organismos Benéficos del Departamento de Agronomía de la Universidad Veracruzana. También consideraremos colocarlo en la web y será un Recuperador de Imágenes Basado en Contenido para cualquier base de datos siempre y cuando dicha base de datos cumpla con las especificaciones requeridas, cualquier usuario podrá subir su base de datos y utilizar el sistema, lo que significa que este sistema podrá ser aplicado a una gran variedad problemas similares.

Referencias

1. Ying Liu, Dengsheng Zhang, Guojun Lu, Wei-Ying Ma: A survey of content-based image retrieval with high-level semantics. Pattern Recognition Society, accepted 28 April (2006)
2. Sieverding Ewald, Fritz Oehl, Javier Palenzuela, Kurt Ineichen, and Gladstone Alves da Silva: Advances in Glomeromycota taxonomy and classification. IMA Fungus, 2(2): 191-199 (2011)
3. Yong Rui and Thomas S. Huang: Image retrieval: Current techniques, promising directions, and open issues. Journal of Visual Communication and Image Representation, 10:39-62 (1999)
4. Maldonado Bautista José Orlando y Graña Romay Manuel: Estudio de métodos de indexación y recuperación en bases de datos de imágenes. Memoria para obtener al grado de Doctor en Informática.
5. Maldonado Méndez Carolina Gabriela y Ríos Figueroa Homero: Recuperación de Información Visual. Tesis de Maestría, Septiembre (2001)
6. Sawhney Harpreet S. and Hafner James L.: Efficient Color Histogram Indexing. IEEE (1994)
7. Tamura, H., S. Mopi, and T. Yamawaki: Textural features corresponding to visual perception. IEEE Transactions on Systems, Man, and Cybernetics, SMC-8(6): p. 460-473 (1978)
8. Hu, M.K.: Visual pattern recognition by moments invariants. IRE Trans. Information Theory, 8: 179-87 (1962)
9. Flusser J. and Suk T.: Pattern recognition by affine moment invariants. Pattern Recogn., vol.26, pp.167-174 (1993)
10. Zhang Dengsheng: Improving Image Retrieval Performance by Using Both Color and Texture Features.
11. Data Base by Peter Shimeld ggpws@cc.newcastle.edu.au for the Pollen Laboratory at the The University of Newcastle, Callaghan, NSW, Australia, NSW, Australia.

Segmentación de imágenes para el análisis de lesiones en la piel: un estudio

Juan García y Salvador Venegas

Instituto Tecnológico y de Estudios Superiores de Monterrey,
Campus Estado de México, México
{A00465790, svenegas}@itesm.mx

Resumen Los recientes avances en visión computacional y en segmentación de imágenes han hecho posible ayudar al diagnóstico médico para una evaluación más precisa de los pacientes. Tal es el caso de las condiciones cutáneas en las que las lesiones de la piel constituyen un indicio visual para el diagnóstico y tratamiento. Por lo tanto, el problema de la segmentación de lesiones en la piel a partir de imágenes adquiridas tanto tradicionalmente como por medio de equipos especiales se ha convertido en un tema de investigación interesante. En este trabajo, los enfoques más comunes y eficaces para atacar dicho problema se presentan de una manera organizada con el fin de identificar sus características y provocar la reflexión sobre nuevas formas de aplicar, combinar o ampliar los métodos previamente explorados.

Palabras clave: visión computacional, segmentación de imágenes, diagnóstico médico, lesiones en la piel, estudio.

1. Introducción

Gracias a los constantes avances en el campo, la visión por ordenador ha hecho posible atacar ciertos problemas que antes eran muy difíciles debido a limitaciones de recursos computacionales. Un área que se ha beneficiado particularmente de esto es la medicina.

Muchas condiciones cutáneas se diagnostican mediante el examen visual de la zona afectada del cuerpo, ya sea directamente (por ejemplo, con las lesiones de piel tales como el melanoma) o indirectamente a través de imágenes obtenidas por una dermatoscopia o cámara térmica (quemaduras).

Aunque los criterios para el diagnóstico de enfermedades suelen estar bien establecidos, puede ser difícil (incluso para especialistas bien entrenados) detectar con precisión las anomalías con un buen porcentaje de certeza que es por lo general alrededor del 80% en adelante. Por desgracia, los especialistas para algunas enfermedades y condiciones son simplemente insuficientes incluso en las grandes ciudades. Por lo tanto, el diagnóstico inicial generalmente no se lleva a cabo por dicho personal y la precisión de la estimación cae dramáticamente lo que puede resultar en un tratamiento y evolución pobre.

Afortunadamente, muchos de estos diagnósticos visuales pueden ser ayudados por software automatizado que en la mayoría de los casos no va a sustituir o competir con la experiencia de un especialista, pero puede hacer viable tanto para el especialista y no especialista por igual, el aumentar drásticamente su evaluación de un paciente dado.

Esto hace de la visión por ordenador, una herramienta muy valiosa para el diagnóstico médico. El papel que juega la visión computacional en el campo es por lo general (aunque no exclusivamente) el de segmentar con precisión las partes de una imagen para su evaluación cuantitativa. Como es de esperar, los tipos de imágenes que se utilizan pueden variar considerablemente y las técnicas utilizadas para hacerlo también.

El objetivo de este trabajo es presentar de manera organizada las diferentes técnicas de segmentación de imágenes que se han utilizado en el diagnóstico de lesiones de piel con el fin de identificar las características que los hacen más adecuados para la tarea.

2. Lesiones de la piel y otras condiciones cutáneas

Las condiciones cutáneas son las que afectan al sistema integumentario humano (compuesto por la piel, el cabello, las uñas y las glándulas relacionadas). Miles de enfermedades de la piel se han descrito, aunque sólo para un puñado de ellos los pacientes buscan tratamiento médico. Las lesiones cutáneas constituyen una indicación visual de la mayoría de estas condiciones en las que el diagnóstico se puede hacer mediante el análisis de su localización, los síntomas, el acomodo, la duración, la morfología y el color. A medida que estos parámetros son fácilmente cuantificables, un diagnóstico ayudado por ordenador es útil para eliminar el error humano en relación con la estimación de los porcentajes de estas cantidades.

Antes de comenzar cualquier evaluación, los médicos deben localizar visualmente y examinar las lesiones de piel del paciente. Este proceso, que es trivial para un ser humano, es el más difícil de realizar para el sistema computacional sin importar los tipos de imágenes utilizados (que son obtenidas desde una simple cámara digital hasta de equipos sofisticados). Aquí es donde la visión por ordenador se vuelve crucial.

3. ¿Qué es la segmentación de imágenes?

La segmentación es la operación de bajo nivel que toma una imagen e intenta particionarla mediante la determinación de conjuntos disjuntos, regiones homogéneas o, alternativamente, mediante la búsqueda de bordes que indican la presencia de un objeto [1]. Este es el proceso que el cerebro humano debe llevar a cabo con el fin de determinar las diferentes partes de una escena.

3.1. Imágenes a color

La segmentación de imágenes no se limita a imágenes de color. Para algunos algoritmos, una imagen en blanco y negro sería suficiente para obtener información significativa. En el diagnóstico médico sin embargo, su color aporta información muy valiosa ya que los médicos analizan imágenes en color en la mayoría de los casos. Las ligeras variaciones de tonalidad pueden ser cruciales para un diagnóstico correcto. Las imágenes en blanco y negro son generalmente limitadas a los valores en el rango [0-255], el cual está muy lejos de la gama de la visión humana. Por lo tanto, se necesita una mejor representación por lo que se prefiere el color.

Aunque el uso de imágenes en color aumenta sustancialmente la complejidad de los cálculos, el beneficio que se refleja en la precisión bien vale la pena. La potencia de cálculo adicional que se necesita viene del número de combinaciones disponibles en la forma en la que se representa el color mediante tuplas de números (por lo general tres componentes). Esto se llama un espacio de color.

4. Espacio de color

El Color, y más precisamente la luz, es un tipo de radiación visible para el ojo humano. Hay varias maneras de representar esta señal analógica en una imagen digital. Por mucho el modelo más común es RGB en el que cada canal representa el monto de rojo, verde y azul que representa un píxel. Sin embargo, este espacio de color RGB es rara vez una buena elección cuando se trata de segmentación de imágenes. Otros numerosos espacios cromáticos existen como LUV, YUV, HSI, HSV y YIQ. Un amplio estudio sobre las propiedades y efectos de espacio de color en la segmentación de imágenes se puede encontrar en [2].

Los trabajos acerca de segmentación de lesiones de piel comúnmente enfatizan el espacio de color utilizado y la razón de su elección. Aunque RGB proporciona suficiente información de color, su representación no es tan buena para lesiones en la piel como se ve en [3]. El algoritmo en [4] fue probado con 25 combinaciones de canales de color de los diferentes espacios de color. Sus mejores resultados se lograron por combinaciones del espacio de color XYZ. En [5], llevan a cabo la segmentación para una piel sana donde encontraron el espacio de color HSV como el más adecuado.

5. Clasificación y pasos de los algoritmos de segmentación de imágenes

En términos generales, todos los algoritmos de segmentación de imágenes constan de tres etapas: pre-procesamiento, procesamiento y post-procesamiento. Sin embargo, los pasos primero y tercero no siempre son necesarios dependiendo del enfoque y la robustez del algoritmo.

Pre-procesamiento es el paso de la eliminación de ruido en la imagen original. Dicho ruido no sólo se genera por la resolución del equipo utilizado para adquirir

la imagen, sino también por las propiedades naturales del sistema integumentario tales como el cabello [4].

La etapa de post-procesamiento es similar a la de pre-procesamiento, pero se utiliza para eliminar pequeños errores en la segmentación resultante (tales como píxeles aislados) a través de operaciones tales como la morfología matemática. Este suele ser necesario cuando el proceso no es lo bastante robusto o considera sólo una propiedad de las imágenes como el color, tal como se muestra en [3].

Al ser una parte integral de la visión computacional, la segmentación de imágenes ha sido ampliamente estudiada. El comparar la precisión de los algoritmos de segmentación no es tan sencillo. Algunos experimentos comparativos se proponen en [6]. En cuanto a su clasificación, Lucchese y Mitra proporcionan una clasificación exhaustiva de métodos de segmentación de imágenes en color en [1], dividida en tres categorías principales: técnicas basadas en atributos, técnicas basadas en la imagen y técnicas basadas en Física. Uno puede ver esta clasificación como una pirámide donde cada categoría va siendo más robusto que la anterior.

Las técnicas basadas en atributos se refieren a aquellos que tratan el problema de la segmentación de la imagen como la de la búsqueda de clústeres en un espacio de color o un espacio generado por sus atributos, sin tener en cuenta otras pistas como las relaciones espaciales. Agrupamiento general, Agrupamiento adaptable k-means y la umbralización del histograma se consideran en esta categoría.

Las técnicas basadas en la imagen son la que se basan en el hecho de que los puntos de un mismo objeto tienden a estar espacialmente cerca. Dividir y mezclar, crecimiento de regiones, técnicas de clasificación basados en bordes y redes neuronales son parte de esta categoría.

Las técnicas basadas en física van un paso más allá al tener en cuenta el ruido en las imágenes mediante el análisis de cómo la luz interactúa con los materiales representados. Modelos de reflexión y otros métodos estado del arte representan esta categoría que aún ha sido poco explorada en comparación con las otras.

Los algoritmos en la siguiente sección de este documento (en adelante, *clásicos*) caen en una de las dos primeras categorías propuestas en [1]. Enfoques heurísticos y los menos frecuentes (en adelante, *alternativos*) les siguen en su propia sección.

6. Segmentación clásica de imágenes de lesiones cutáneas

Se han propuesto varios algoritmos para la segmentación de lesiones cutáneas en las dos últimas décadas. Algunos de los más destacados y estudiados se discuten en esta sección.

Segmentar por el color solamente es, por lo general, el primer intento al segmentar una imagen (técnicas basadas en atributos). Un ejemplo de tales algoritmos se presenta en [3]. Dos mejoras en la imagen se aplican: la normalización del color y mejora del contraste. Estos ayudan a la segmentación pues se hace más fácil encontrar bordes y el ruido de la imagen se reduce. Después, un algoritmo ISODATA se aplica iterativamente para determinar el valor de umbral óptimo entre lesión de la piel y la piel saludable. La precisión se considera buena, aunque

visualmente, los resultados no se presentan suficientemente convincentes. Otro ejemplo de segmentación básica se puede encontrar en [7].

La necesidad de enfoques más robustos se exploró con las técnicas basadas en la imagen. En [4] y [8], la segmentación se realiza por medio de un análisis del espacio de color y el agrupamiento basado en umbralización de histograma para imágenes dermatoscópicas. Su algoritmo toma en cuenta la existencia de vello en las imágenes, que se elimina en la etapa de pre-procesamiento mediante operaciones morfológicas en las direcciones vertical, horizontal y diagonal y la interpolación de los píxeles que faltan después en sus alrededores. Un filtro de promedio (*mean*) se aplica entonces para eliminar el ruido del vello sobrante. La etapa de segmentación se realiza mediante la clasificación de cada píxel de una imagen en blanco y negro (con intensidad ajustada) en dos clases mediante el método de Otsu. Por último, se realiza un análisis de componentes conectados para eliminar los objetos adicionales, tales como marcas de los médicos en los alrededores. Las dos áreas más importantes son tomadas como lesión y la piel y el resto se descartan. Según sus pruebas, un 98 % de precisión se logró al comparar sus resultados con los de los médicos expertos.

Otro ejemplo de agrupamiento en segmentación de imágenes se muestra en [9] utilizando K-means, un algoritmo clásico.

Keke et al. [5] toman un enfoque diferente segmentando la piel sana en lugar de las lesiones. La Aritmética de Fuzzy-c-means mejorada se utiliza como el algoritmo para la segmentación donde toman cada píxel e intentan clasificarlo para un conjunto de centroides de grupo. Para evitar el ruido esperado en la imagen, una mejora se aplica al algoritmo mediante la introducción del concepto de la atracción de vecindario. Con esta adición, su exactitud máxima fue de 94 %, a pesar de que concluyen que este aumento de la precisión se refleja también en un aumento en el tiempo de procesamiento.

Wong et al. utilizan un algoritmo estocástico e iterativo de fusión de regiones en [10] con el fin de reducir los problemas que surgen de la adquisición de las imágenes con las cámaras convencionales. Cada píxel de la imagen se marca como perteneciente a una región única y estas regiones son posteriormente fusionadas con otras regiones de una manera estocástica basada en una función de similitud de regiones. Los resultados muestran que este método es más robusto a problemas de iluminación que otros métodos comparables.

Para algunas lesiones de la piel, los bordes proporcionan tanto una forma práctica de segmentar la lesión, e información acerca de la misma. Por lo tanto, la detección de bordes constituye otro enfoque relevante. Un ejemplo se puede encontrar en [11], donde se utiliza un detector de borde LOG para la detección eficaz de los mismos en las imágenes ruidosas. Una imagen RGB de entrada se convierte a una imagen de intensidad en escala de grises mediante la retención del canal de luminancia únicamente. Esta imagen se transforma entonces en una imagen binaria utilizando valores de umbral y es aquí donde se aplica el detector de borde LOG. Los resultados reportados son satisfactorios, incluso en la presencia de pelo y ruido añadido. Este método puede no funcionar con todo tipo de lesiones en la piel, pero puede ser una herramienta útil para otros

enfoques. Un trabajo similar usando detección de los bordes se presenta en [12] donde la imagen también se transforma en una imagen binaria pero en este caso se utilizan los coeficientes de wavelets con el fin de obtener una imagen mejorada con un mayor contraste. Morfología de operadores es utilizada entonces para segmentar esta imagen. Esta técnica muestra resultados prometedores con un bajo porcentaje de error en los bordes. Detección de bordes para las lesiones no melanocíticas se trata en [13]. Detección de bordes a través de algoritmos de serpiente (contorno activo) se analiza y mejora (incorporando un término promedio de campo a la función objetivo estándar en GVF) en [14].

Los trabajos previos no hacen uso de ningún tipo de conocimiento a priori. En [15], el conocimiento a priori se utiliza para un enfoque de clasificación de píxeles basado en ejemplares para imágenes dermatoscópicas. El objetivo de este método de segmentación es clasificar cada bandeja de histograma como una lesión o piel al compararlos con los ejemplares vecinos identificados a través de correspondencia de histogramas chi-cuadrada. Se logra suavidad espacial con la suposición de que píxeles cercanos probablemente pertenecen a la misma clase. Los autores proponen como ventajas una alta precisión y la eficiencia computacional, aunque vale la pena señalar que el requisito de la base de datos previamente categorizada puede constituir un serio obstáculo.

Como se mencionó anteriormente, las imágenes en color son por lo general, favorecidas sobre las imágenes en blanco y negro por proporcionar más información. Sin embargo, una combinación de ambos también se puede utilizar como en [16] cuando resultados menos estrictos son aceptables. En este trabajo, una imagen en color se recibe como entrada para la segmentación. Esta se transforma en el espacio de color HSB que lleva la mayor parte de la información para el color gris en el componente B. Con la imagen en escala de grises, se utiliza un enfoque estadístico para encontrar los valores que mejor capturan la región de interés de la lesión. El valor promedio de brillo y el histograma en la parte de la lesión se calculan para obtener el intervalo de umbralización correspondiente a 30% del valor relativo del canal gris. Como se observa en los resultados presentados, este enfoque no es suficiente para segmentar con precisión las lesiones de la piel en la imagen. Sin embargo, no debe ser desechado ya que puede ser un posible complemento para uno de los métodos anteriores.

7. Segmentación alternativa de imágenes para lesiones cutáneas

A pesar de que la complejidad de los algoritmos mencionados es por lo general P (es decir, ejecutados en un número polinómico de pasos de acuerdo a la entrada), el gran número de píxeles en algunas de las imágenes producidas por los dispositivos modernos hacen que estos algoritmos sean menos prácticos para la evaluación inmediata, especialmente cuando son llevados a dispositivos móviles. Por esta razón, los enfoques heurísticos que se han aplicado en otras áreas de diagnóstico médico de visión por computador se presentan aquí.

Un enfoque de sistemas multi-agente para imágenes de tomografía computarizada es presentado en [17]. El sistema consta de varios agentes que tienen una sub-imagen y tratan de marcar cada píxel de su región por medio de reconocimiento a priori. Agentes moderador comprueban después el resultado de los agentes anteriores para producir un resultado final. La negociación se pondrá sólo a disposición de los agentes de trabajo que no poseen suficiente información para tomar una decisión. Este enfoque ha sido probado con pequeñas imágenes (512x512 píxeles) y encontró que los resultados estuvieron muy cerca de los datos segmentados manualmente (en torno al 90%). Como una ventaja, este método no requiere ninguna interacción aunque el tiempo de cálculo es alta incluso para imágenes pequeñas.

Los algoritmos genéticos se han aplicado también a algoritmos de segmentación para imágenes médicas. Las imágenes ruidosas que se utilizan por lo general hacen la función objetivo más compleja, multi-modal y discontinua [18]. Esto hace a los algoritmos genéticos especialmente útiles para dicha tarea. Las funciones multi-objetivo también son útiles pues varios parámetros se pueden usar para segmentar una imagen. Un estudio sobre la aplicación del AGs a las imágenes médicas se encuentra en [18].

La estrategia evolutiva, una técnica de optimización de búsqueda basada en el azar, se prueba en [19]. De manera similar a los algoritmos genéticos, éste emplea una forma elíptica como la base para segmentar la lesión. Esta forma es modificada cada generación para buscar una mejor salida. De acuerdo con sus resultados, este método es robusto y automático. Es claro, sin embargo que se necesita más información de la función de aptitud para evaluar correctamente las lesiones de morfología poco común (que contiene huecos, por ejemplo).

7.1. Paralelización

Con la reciente disponibilidad de GPUs de propósito general para el mercado de consumo, la mayoría de los algoritmos de visión por computador pueden ser revisados una vez más para que tomen ventajas del computo en paralelo que proporcionan estos dispositivos. Los algoritmos mencionados en este documento no son la excepción, al contrario, son candidatos naturales para dicha opción.

El sistema multi-agente en [17] es un claro ejemplo, donde cada agente puede hacer su trabajo de forma paralela sin afectar a los demás. Del mismo modo, cada algoritmo que aplica la misma operación para cada píxel (o grupo de píxeles) es propenso a la paralelización. Tal es el caso de los algoritmos de clasificación de técnicas basados en atributos y la mayoría de los que pertenecen a las técnicas basadas en la imagen.

8. Comparación de los algoritmos principales

A continuación se muestra una tabla con el resumen de los principales algoritmos mencionados en el presente trabajo, incluyendo el tipo de imágenes en las que fueron aplicados.

Tabla 1. Algoritmos para la segmentación de lesiones en la piel.

Técnicas de Segmentación	
Clásicas	
Técnica	Tipo de imagen
Algoritmo ISODATA	Imágenes Dermatoscópicas de Melanoma
Umbralización del Histograma	Imágenes Macroscópicas de Melanoma
Umbralización con el Método de Otsu	Imágenes Dermatoscópicas Varias
Aritmética de Fuzzy-C-Means	Imágenes Dermatoscópicas de Melanoma
Agrupamiento K-Means	Imágenes Macroscópicas de Lupus
Fusión de Regiones Estocástica	Imágenes Macroscópicas de Melanoma
Detector de Bordes LOG	Imágenes Dermatoscópicas de Melanoma
Coefficientes <i>Wavelet</i> (Bordes)	Imágenes Dermatoscópicas Varias
Detección Multifase de Bordes	Imágenes Dermatoscópicas Varias
Modelo de Serpiente Mejorado (Bordes)	Imágenes Dermatoscópicas de Melanoma
Clasificación de Píxeles por Ejemplares	Imágenes Dermatoscópicas Varias
Análisis Estadístico de Imagen	Imágenes Dermatoscópicas de Melanoma
Alternativas	
Técnica	Tipo de imagen
Sistema Multi-agente	Tomografías por Computadora
Algoritms Genéticos	Resonancia Magnética y Tomografías
Estrategia Evolutiva	Microscopía de Melanoma

9. Conclusión

Sin duda, la medicina se ha beneficiado enormemente de los avances en la visión por computador y más específicamente, en los avances en algoritmos de segmentación de imágenes. Virtualmente cualquier enfermedad que se puede diagnosticar visualmente se puede automatizar en cierta medida mediante un programa de visión por computador que se convierte en una herramienta muy valiosa para los médicos.

En este trabajo, tanto los métodos de segmentación clásicos como alternativos fueron discutidos como un medio para identificar las características que los diferencian y hacen poderosos para ciertas tareas. Métodos incluidos en las categorías basadas en atributos y basados en la imagen fueron etiquetados como clásicos. Enfoques más heurísticos fueron referidos como alternativos.

Este documento sólo recoge el trabajo en un área muy específica de la visión por computador médica que es la segmentación de imágenes. Sin embargo, todavía hay mucho más procesamiento una vez que la zona afectada se ha identificado como la clasificación y estimación de tratamiento. Al conectar cada etapa del proceso se obtienen sistemas de visión por computador y herramientas CAD que ya pueden ser utilizados en hospitales y centros de atención médica. Un amplio estudio de estos sistemas se encuentra en [20].

La segmentación de Imágenes para el diagnóstico médico es en sí mismo un enorme tema que, a pesar de que ha sido investigado ampliamente, aún presenta interesantes retos para la comunidad científica. En este trabajo se fomenta el análisis de los algoritmos utilizados previamente y sus beneficios aplicados en ciertas enfermedades para atacar nuevos problemas no sólo en condiciones cutáneas, sino también en otras enfermedades que aún no se han explorado.

Referencias

1. Yz, L.L., Y, S.K.M.: Color image segmentation: A state-of-the-art survey (2000)
2. Kwok, N., Ha, Q., Fang, G.: Effect of color space on color image segmentation. In: Image and Signal Processing, 2009. CISP '09. 2nd International Congress on. (oct. 2009) 1 –5
3. Schaefer, G., Rajab, M., Celebi, M., Iyatomi, H.: Skin lesion extraction in dermoscopic images based on colour enhancement and iterative segmentation. In: Image Processing (ICIP), 2009 16th IEEE International Conference on. (nov. 2009) 3361 –3364
4. Garnavi, R., Aldeen, M., Celebi, M.E., Bhuiyan, A., Dolianitis, C., Varigos, G.: Skin lesion segmentation using color channel optimization and clustering-based histogram (2009)
5. Keke, S., Peng, Z., Guohui, L.: Study on skin color image segmentation used by fuzzy-c-means arithmetic. In: Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on. Volume 2. (aug. 2010) 612 –615
6. Kang, W.X., Yang, Q.Q., Liang, R.P.: The comparative research on image segmentation algorithms. In: Education Technology and Computer Science, 2009. ETCS '09. First International Workshop on. Volume 2. (march 2009) 703 –707
7. Lee, T., Ng, V., McLean, D., Coldman, A., Gallagher, R., Sale, J.: A multi-stage segmentation method for images of skin lesions. In: Communications, Computers, and Signal Processing, 1995. Proceedings. IEEE Pacific Rim Conference on. (may 1995) 602 –605
8. Humayun, J., Malik, A., Kamel, N.: Multilevel thresholding for segmentation of pigmented skin lesions. In: Imaging Systems and Techniques (IST), 2011 IEEE International Conference on. (may 2011) 310 –314
9. Kesede, R., Lee, L., Bassani, J.: Automatic identification of discoid lupus erythematosus. In: Biosignals and Biorobotics Conference (BRC), 2011 ISSNIP. (jan. 2011) 1 –6
10. Wong, A., Scharcanski, J., Fieguth, P.: Automatic skin lesion segmentation via iterative stochastic region merging. Information Technology in Biomedicine, IEEE Transactions on **PP**(99) (2011) 1
11. Jaseema Yasmin, J., Mohamed Sathik, M., Beevi, S.: Robust segmentation algorithm using log edge detector for effective border detection of noisy skin lesions. In: Computer, Communication and Electrical Technology (ICCCET), 2011 International Conference on. (march 2011) 60 –65
12. Sarrafzade, O., Baygi, M., Ghassemi, P.: Skin lesion detection in dermoscopy images using wavelet transform and morphology operations. In: Biomedical Engineering (ICBME), 2010 17th Iranian Conference of. (nov. 2010) 1 –4
13. Norton, K.A., Iyatomi, H., Celebi, M., Schaefer, G., Tanaka, M., Ogawa, K.: Development of a novel border detection method for melanocytic and non-melanocytic dermoscopy images. In: Engineering in Medicine and Biology Society (EMBC),

- 2010 Annual International Conference of the IEEE. (31 2010-sept. 4 2010) 5403–5406
14. Zhou, H., Schaefer, G., Celebi, M., Iyatomi, H., Norton, K., Liu, T., Lin, F.: Skin lesion segmentation using an improved snake model. In: Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE. (31 2010-sept. 4 2010) 1974–1977
 15. Zhou, H., Rehg, J., Chen, M.: Exemplar-based segmentation of pigmented skin lesions from dermoscopy images. In: Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on. (april 2010) 225–228
 16. Sforza, G., Castellano, G., Stanley, R., Stoecker, W., Hagerty, J.: Adaptive segmentation of gray areas in dermoscopy images. In: Medical Measurements and Applications Proceedings (MeMeA), 2011 IEEE International Workshop on. (may 2011) 628–631
 17. Chitsaz, M., Seng, W.C.: A multi-agent system approach for medical image segmentation. In: Future Computer and Communication, 2009. ICFCC 2009. International Conference on. (april 2009) 408–411
 18. Maulik, U.: Medical image segmentation using genetic algorithms. Information Technology in Biomedicine, IEEE Transactions on **13**(2) (march 2009) 166–173
 19. Situ, N., Yuan, X., Zouridakis, G., Mullani, N.: Automatic segmentation of skin lesion images using evolutionary strategy. In: Image Processing, 2007. ICIP 2007. IEEE International Conference on. Volume 6. (16 2007-oct. 19 2007) VI–277–VI–280
 20. Maglogiannis, I., Doukas, C.: Overview of advanced computer vision systems for skin lesions characterization. Information Technology in Biomedicine, IEEE Transactions on **13**(5) (sept. 2009) 721–733

Autenticación de documentos por medio de la extracción de características de las fibras del papel utilizando Wavelets

Rodolfo Ibarra-Orozco¹, Juan Carlos López-Pimentel¹, Miguel González-Mendoza² y Víctor Ramos-Fon Bon¹

¹ Universidad Politécnica de Chiapas, México

{rodolfo.ibarra, jcarlos, vramos}@upchiapas.edu.mx

² Instituto Tecnológico y de Estudios Superiores de Monterrey, México
mgonza@itesm.mx

Resumen Este artículo describe una primer propuesta para autenticar un documento físico mediante una metodología para extraer, de una pequeña porción de una hoja de papel, características relevantes de las fibras que la componen. Esta extracción de características se realiza empleando la transformada en wavelets. Se demuestra que los coeficientes de las wavelets son útiles para extraer, después de un preprocesamiento de la imagen, esas características relevantes. Posteriormente, al utilizar un algoritmo de compresión, se logra reducir el número de coeficientes de wavelets sin perder la información importante. Los resultados obtenidos muestran que es factible realizar la autenticación de documentos usando la metodología propuesta, habiendo aún la posibilidad de mejorar cada uno de los pasos.

Palabras clave: wavelets, autenticación de documentos

1. Introduction

La falsificación de documentos físicos es un problema actual de gran repercusión. Esto ha propiciado la creación de propuestas para validar la autenticidad de dichos documentos. No obstante, cada propuesta implica costos muy elevados para su realización. Por ejemplo, [5], sólo para el Estado de México, para la elección de nuevos ediles y diputados locales, se calcula que si el IFE implementa una nueva medida de seguridad en el papel de 22 millones de boletas, se tendría un gasto adicional de 22 millones de pesos.

El escenario descrito ha inspirado el presente proyecto, el cual se centra en el desarrollo de una aplicación que 1) permita validar la veracidad de la información que contiene un documento, al mismo tiempo que 2) indique si el documento que se está cotejando está impreso la hoja en la que se imprimió originalmente. El proceso se realiza mediante un sistema de impresión, es decir, el sistema debe, al momento de ser impreso el documento, también imprimir una marca con información sobre las características del papel.

Se identificaron trabajos similares, entre los que destacan los siguientes: Buchanan y Cowburn, [3], proponen un sistema que mide la textura de una de una página para identificar un documento usando un microscopio de alto poder. El precio de este equipo es prohibitivo para muchos usuarios.

Zhu, [1], identifica "aleatoriedad no repetitiva existente en el proceso de impresión". La desventaja de este proceso es que primero es necesario imprimir el documento para después analizarlo e imprimir después la marca.

Clarkson, [2], propone un sistema que utiliza un escáner para obtener características relevantes de una hoja de papel e insertar esas características en una marca. La extracción de características las realiza mediante algoritmos estadísticos. En nuestro caso ocuparemos algoritmos de análisis de señales en tiempo-frecuencia y algoritmos de inteligencia artificial.

El artículo está organizado como sigue: primero se presenta una introducción, después, una descripción básica de los elementos a utilizar en la metodología. La sección 4 describe la metodología utilizada y la sección 6 los resultados obtenidos y el trabajo futuro.

2. Transformada en Wavelets

La mayoría de las señales de fenómenos del mundo real son no estacionarias. Ejemplos de estas son las señales generadas por electroencefalogramas, electroculogramas, voz, sonidos de animales, etc. Estas señales varían con el tiempo su intensidad y su contenido en frecuencia. Así, si se quiere saber en que momento (tiempo) ocurrió una frecuencia, se necesitan otras técnicas para el análisis.

En esta sección, se describen herramientas matemáticas para la descripción de señales no estacionarias. Este tipo de análisis es llamado análisis en tiempo-frecuencia. En particular, describimos el uso de la transformada en Wavelets. Para una referencia más amplia se recomienda consultar [7] y [4].

2.1. La transformada en wavelets continua

Para analizar estructuras de la señal de muy diferentes tamaños, es necesario utilizar átomos en tiempo-frecuencia con diferentes soportes en el tiempo. Una wavelet es una función $\psi \in L^2(\mathbb{R})$ con media cero:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (1)$$

La transformada en wavelets continua (Continuous Wavelet Transform, CWT) se basa en la convolución de una wavelet madre ψ que es dilatada (s) y defasada (τ), (ecuación 2) para tener una aproximación y descomposición de la señal original. La wavelet es desplazada a lo largo de la señal y para cada posición el espectro es calculado. Luego este proceso es repetido muchas veces con una ventana ligeramente más corta o larga en cada nuevo ciclo. Finalmente se obtiene una colección de representaciones tiempo-frecuencia de la señal a diferentes resoluciones o escalas (s), (ecuación 3).

$$\psi_s^\tau(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) \quad (2)$$

$$C(s, \tau) = \langle f, \psi_{s,t} \rangle = \frac{1}{\sqrt{|s|}} \int f(t) \psi\left(\frac{t-\tau}{s}\right) \quad (3)$$

Esta transformación es en teoría infinitamente redundante, pero puede ser útil en el reconocimiento de ciertas características de la señal.

2.2. La transformada en wavelets discreta y multiresolución

En una transformada de wavelets discreta, una wavelet es trasladada y dilatada únicamente por valores discretos. Frecuentemente esta dilatación es a una potencia de 2 (llamada diádica). Así, la wavelet es utilizada de la forma:

$$\psi_s^{2^k t+l} \quad (4)$$

donde k y l son números enteros.

Con la teoría de multiresolución, Stéphane Mallat relacionó las wavelets ortogonales con los filtros utilizados en procesamiento de señales. En este método se utiliza una nueva función, la función de escala, la cual produce una serie de imágenes de la señal, cada una diferente por un factor de dos de la señal previa. En una dirección, estas imágenes sucesivas aproximan la señal con cada vez mayor precisión, acercándose a la original. En la otra dirección, se acercan a cero, conteniendo cada vez menor información.

La función de las wavelets en este método es la de obtener la diferencia de información entre dos resoluciones.

A las señales obtenidas al aplicarles la función de escala se les llama aproximaciones y a las obtenidas al aplicarles la wavelet se les llama detalles.

La multiresolución puede aproximar cualquier función sin redundancia y con precisión arbitraria si son cumplidas las siguientes cuatro condiciones:

- La función de escala debe ser ortogonal a sus traslados por enteros.
- La señal, descompuesta a una resolución dada j , contiene toda la información de esa misma señal descompuesta a una resolución menor $j - k$.
- La función 0 es el único objeto común en todos los espacios V_j .
- Cualquier señal puede ser aproximada con una precisión arbitraria.

La multiresolución es descrita primero por los subespacios V_j y W_j . Los espacios de escala V_j . W_j es la diferencia entre V_j y V_{j+1} . La suma de V_j y W_j es V_{j+1} . Cada V_j está contenido en el siguiente subespacio V_{j+1} . Una función en un subespacio está en todos los subespacios más altos (finos).

$$V_0 \subset V_1 \cdots V_j \subset V_{j+1} \subset \quad (5)$$

Una función $f(t)$ en el espacio completo tiene una pieza en cada subespacio. Esas piezas contienen más y más información de la información completa en $f(t)$.

La pieza en V_j es $f_j(t)$. Un requerimiento en la secuencia de subespacios es la completos:

$$f_j(t) \rightarrow f(t) \text{ as } j \rightarrow \infty \quad (6)$$

Visto desde el punto de vista en que

$$V_j \oplus W_j = V_{j+1} \quad (7)$$

Cada función en V_j es la suma de dos partes ortogonales, f_j en V_j y Δf_i en W_j .

El espacio W_j contiene términos de grado $j+1$. Estos son ortogonales a todos los términos de grado $\leq j$.

$$V_0 \oplus W_0 = V_1 \quad \text{y} \quad V_1 \oplus W_1 = V_2 \quad (8)$$

De donde se tiene que:

$$V_0 \oplus W_0 \oplus W_1 = V_2 \quad (9)$$

Agregando más detalles se tiene:

$$V_0 \oplus W_0 \oplus W_1 \oplus \dots \oplus W_j = V_{j+1} \quad (10)$$

para las funciones en estos subespacios, esta ecuación es:

$$f_0(t) + \Delta f_0(t) + \Delta f_1(t) + \dots + \Delta f_j(t) = f_{j+1}(t) \quad (11)$$

La construcción de wavelets ha tenido éxito encontrando V_j primero, es decir, la función de escala $\psi(t)$. Las translaciones de esta wavelet $\psi(t-k)$ caen en V_0 . Reescalando por 2^j se obtiene V_j . Entonces los espacios de la wavelet W_j son las diferencias entre V_{j+1} y V_j . Las funciones en W_j son los detalles en la escala j .

De manera similar para los bancos de filtros, primero se diseña el filtro pasa bajas y después se diseña el filtro pasa altas, que son análogos a la función de escala y a la función wavelet, respectivamente.

2.3. Algoritmo EZW

El algoritmo EZW (Embedded Zero-Tree Wavelet), [6] fue presentado en el artículo de Shapiro. La idea principal de la compresión EZW es la explotación de similitudes en diferentes escalas de la transformada en wavelets de una imagen. Este algoritmo es considerado el primer codificador de wavelets realmente eficiente. Su rendimiento se basa en la similitud entre sub-bandas y un esquema de aproximaciones sucesivas. Los coeficientes en diferentes sub-bandas del mismo tipo representan la misma locación espacial, en el sentido que un coeficiente en una escala corresponde con cuatro en un nivel anterior. Esta conexión puede ser utilizada recursivamente con estos cuatro coeficientes y sus correspondientes de los niveles más bajos, de forma que se pueden definir árboles de coeficientes. En las imágenes la mayor parte de la energía tiende a concentrarse en los niveles más altos de descomposición, por lo que se puede esperar que entre un coeficiente se encuentre más cerca al nodo raíz, su magnitud será mayor, así que si un nodo en un árbol de coeficientes es menor a un umbral, es muy probable que sus coeficientes descendientes serán también pequeños.

3. Extracción de características basada en la Transformada Discreta de Wavelets

La Transformada Discreta en Wavelets obtiene tanto la información espacial y frecuencial de una señal. Esta transformada analiza la imagen utilizando un filtrado pasa bajas para obtener características burdas de la señal (aproximaciones) y un filtrado para altas para adquirir las características finas de la señal (detalles). Tal descomposición es realizada recursivamente en los coeficientes de la aproximación obtenidos en cada nivel, hasta que las iteraciones necesarias son alcanzadas. Una imagen es representada como una matriz en escala de grises donde cada elemento de la matriz representa una intensidad en escala de grises de un pixel. Cada pixel que no se encuentra en los bordes de la imagen tiene 8 pixeles adyacentes. Para descomponer una imagen utilizando la TDW se toman en cuenta cuatro direcciones de descomposición correspondientes a 0° C (horizontal, Dh), 45° C (Diagonal, Dd) y 90° C (vertical, Dv).

La descomposición bidimensional con wavelets en J octavas de una imagen discreta representa a la imagen en $3J + 1$ subimágenes.

$$(a_J, \{dh_j, dv_j, dd_j\}_{j=1,J}) \quad (12)$$

donde a_J es una aproximación a baja resolución de la imagen original y dh_j, dv_j, dd_j son los coeficientes que contienen los detalles de la imagen en las tres orientaciones a diferentes escalas 2^j . Los coeficientes de wavelets con una amplitud grande en los detalles corresponden a altas frecuencias verticales (bordes horizontales), altas frecuencias horizontales (bordes verticales) y altas frecuencias en ambas direcciones.

4. Autenticación de documentos utilizando las características de las fibras que componen la hoja de papel

4.1. Metodología

La metodología consiste de dos procesos principales, uno que es la extracción de características de la hoja de papel para el marcado de la hoja y el segundo que es la validación del documento, es decir extraer las características de la hoja de papel y constatar que corresponde a las del marcado realizado en el primer proceso.

Proceso 1. Extracción de características y marcado del papel.

1. Adquisición de la imagen.
2. Aumentar contraste en la imagen.
3. Binarización de la imagen.
4. Detección de bordes.
5. Descomposición de la imagen utilizando la transformada en wavelets.
6. Reducción del número de coeficientes utilizando el algoritmo EZW.

7. Extracción de un vector de características.
8. Marcado del papel.

Proceso 2. Validación.

1. Realizar los primeros 7 pasos del proceso 1.
2. Comparación del vector de características con el contenido en el marcado.

Adquisición de la imagen. Se selecciona aleatoriamente una porción de la imagen. En estas pruebas se tomó la imagen con una cámara digital de 8 megapíxeles. Se espera una mejora significativa del proceso al adquirir la imagen con una cámara especializada. En ese caso se seleccionó una porción de 1 cm X 1 cm. Figura 1

Aumentar contraste en la imagen Para el incremento de contraste se aplica una curva s-shape para cada uno de los canales RGB. Figura 2.

Binarización de la imagen Se selecciona un umbral y la imagen se convierte a imagen en blanco y negro.

Detección de bordes Se aplicó el algoritmo del gradiente para la detección de bordes.

Descomposición de la imagen utilizando la transformada en wavelets Se realizó una descomposición a primer nivel utilizando una wavelet Daubechies 2. Figura 3.

Reducción del número de coeficientes utilizando el algoritmo EZW Se emplea el algoritmo EZW para reducir el número de coeficientes en la transformada en wavelets. Se tiene en promedio, en estas imágenes, una compresión del 60%. Este paso se realiza pensando en una extracción de un vector de características que mida todos los coeficientes de la imagen. Figura 4.

Extracción de un vector de características Se divide cada una de las tres subimágenes de los detalles de la descomposición en 16 cuadros de igual tamaño y se calcula la desviación estándar de los coeficientes de cada uno estos cuadros.

Marcado del papel El vector de características y la posición de la parte de la hoja analizada se registra en un código QR.

Comparación del vector de características con el contenido en el marcado Para validar si el marcado en el papel corresponde a las características del papel, se analiza la porción de la hoja que indica el marcado y se extraen, utilizando los pasos descritos, el vector de características. Se mide la similitud entre este vector y el contenido en el marcado. La distancia de Hamming es utilizada como métrica.

5. Resultados

Para medir los resultados obtenidos se extrajeron 100 porciones de hojas de papel diferentes. Una vez realizado la extracción del vector de características de cada porción, se mide la similitud de cada vector de características con los del conjunto completo de 100 vectores (aplicando la distancia de Hamming).



Figura 1. Imagen original de la textura de una hoja de papel

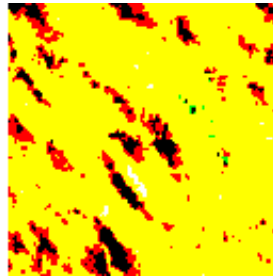


Figura 2. Aumento de contraste en la imagen

Una vez realizadas las comparaciones, se le asigna al vector comparado el vector del conjunto completo que tenga la distancia más corta. En los experimentos realizados, el 85 % de los vectores fueron clasificados correctamente.

6. Conclusiones y trabajo futuro

En este artículo se describe una metodología utilizada para marcar un documento con las características de la fibra del papel y de esta forma validar que dicho documento sea auténtico. Los experimentos realizados muestran que esta es una forma factible de efectuar dicha autenticación. El trabajo futuro es muy extenso y consiste principalmente, a grandes rasgos, de los siguientes puntos:

- Validar la selección de la wavelet a utilizar.
- Implementar un algoritmo de alineación de la imagen.
- Validar el nivel o niveles de descomposición de la transformada en wavelets que nos permita obtener las características principales de las fibras de la hoja de papel.
- Determinar una distancia que permita establecer de forma precisa si las características obtenidas corresponden a las registradas en la marca.
- Establecer una metodología robusta para la validación de resultados.

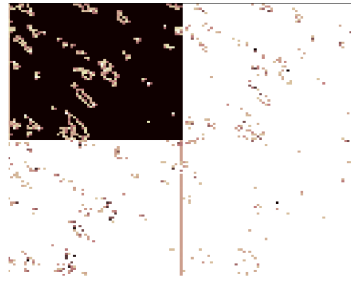


Figura 3. Descomposición de primer nivel de la imagen utilizando db2

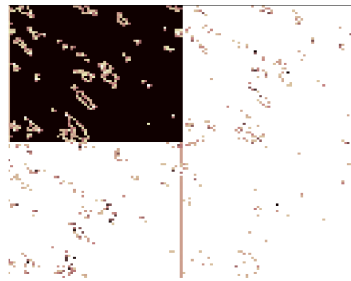


Figura 4. Descomposición de primer nivel de la imagen utilizando db2 y aplicando compresión EZW

Referencias

1. J. Wu B. Zhu and M. S. Kankanhalli. Print signatures for document authentication. *10th ACM Conference on Computer and Communications Security*, 2003.
2. William Clarkson et al. Fingerprinting blank paper using commodity scanners. *IEEE Symposium on Security and Privacy*, 2009.
3. A.-V. Jausovec D. Petit P. Seem G. Xiong D. Atkinson K. Fenton D. A. Allwood J. D. R. Buchanan, R. P. Cowburn and M. T. Bryan. Forgery: ?ngerprinting documents and packaging. *Nature*, 2005.
4. Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.
5. Periódico Milenio. <http://impreso.milenio.com/node/9117434>. 2012.
6. Priyanka Singh and Priti Singh. Design and implementation of ezw and spiht image coder for virtual images. *International Journal of Computer Science and Security (IJCSS)*, 2011.
7. Gilbert Strang and Truong Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Ellesley MA, 1997.

Método robusto para detectar dedos usando profundidad

Isaac Juan Rudomín Goldberg, Jorge Adolfo Ramírez Uresti y Christian Jesús Arzate Cruz

ITESM CEM

Atizapan de Zaragoza, Mexico

rudomin.isaac@gmail.com, juresti@gmail.com, arzate.christian@gmail.com

Resumen Se usa una cámara de profundidad de bajo costo para detectar, de manera robusta, manos y dedos. Nuestro método propuesto está basado en filtros de profundidad y análisis de contorno adaptivo. La mayor parte del enfoque propuesto se programó en CUDA. Seleccionamos CUDA para poder conseguir una aplicación de visión computacional en tiempo real. En las pruebas realizadas conseguimos un 95,17% de efectividad con un 3,2% de falsos positivos.

Palabras clave: detección, manos, dedos, profundidad, Kinect.

1. Introducción

Sistemas de visión computacional puede ser utilizados para crear interfaces convincentes para computadoras. Esto se debe a que es más natural y divertido controlar una computadora usando el movimiento del cuerpo o gestos de manos y dedos. Estas formas de interfaz también se han visto beneficiadas de los precios bajos que encontramos en varios de los componentes que se usan para desarrollar sistemas de visión computacional. Incluso son lo suficientemente baratos para reemplazar interfaces como botones y palancas.

Muchas interfaces han sido desarrolladas para habilitar interacciones naturales con manos y dedos. La mayoría de estos sistemas de visión computacional usan cámaras RGB tradicionales o infrarrojas [1,?,?,?]. Estos métodos encuentran problemas cuando:

- Las manos y el fondo contienen colores similares
- Cambios de iluminación
- Reflejos
- Sombras
- La geometría del fondo cambia

El objetivo del método presentado en este escrito es: desarrollar un sistema de visión computacional en tiempo real que detecte manos y dedos de forma robusta, utilizando una cámara de profundidad. Escogimos este tipo de cámaras porque simplifican todo el sistema de detección. En mayor parte porque no les

afecta los cambios de iluminación, proveen información geométrica directamente, y la tecnología actual las hace asequibles y precisas.

Parte importante de nuestro sistema es su estructura. Se implementó una plataforma que puede ser utilizada para desarrollar distintos tipos de aplicaciones. Además, la mayoría de los algoritmos fueron programados en CUDA para conseguir que nuestro sistema funcione en tiempo real.

Este artículo continúa con la siguiente estructura. En la siguiente sección se describe el hardware que utilizamos y su posición con respecto al usuario. En la sección III, se hace un análisis sobre las metodologías encontradas en el estado del arte para detectar manos y dedos. La sección IV explica los pasos del proceso que sigue nuestro sistema. Los resultados son presentados en la sección V y las conclusiones y trabajo a futuro aparecen en la sección VI.

2. Hardware y descripción de la Escena

Nuestro sistema consiste de una cámara de profundidad de bajo costo (Microsoft Kinect Sensor) el cual provee un mapa de profundidad a una resolución de 640x480 pixeles a 30 Hz. Objetos entre 0.3 y 3.5 m pueden ser vistos por este sensor. Para mantener nuestro sistema lo más general posible optamos por situar la cámara frente al usuario, de forma que podamos verlo de la cintura a la cabeza. También se considera que cuando el usuario desea interactuar con el sistema, este alza sus brazos y muestra sus palmas y dedos como se puede observar en la Figura 3(a).

3. Estado del arte

En la literatura se pueden encontrar distintas formas de detectar manos y dedos. Por ejemplo, podemos ver en [2,?,?,?] que realizan un análisis de contorno, otros métodos usan algoritmos de reconocimiento de objetos [3], emparejamiento de modelos [4,?] o cálculos morfológicos [5] para conseguir su objetivo. Cada método tiene ventajas, desventajas y limitaciones. Ahora analizaremos estos distintos.

3.1. Análisis de contorno

curvatura-k Thiago R. Trigo y Sergio Roberto M. Pellegrino en [1] utilizan el algoritmo curvatura-k para detectar dedos. El primer paso de este método es segmentar las manos y extraer su contorno. Después, para cada uno de los puntos de los contornos encontrados se corre el algoritmo curvatura-k. El algoritmo curvatura-k encuentra el ángulo entre dos vectores (α y β) y toma como entradas; la lista ordenada de puntos que pertenecen al contorno de las manos, y las constantes k y w . La salida de este algoritmo simplemente te dice si el punto en cuestión es parte o no de la curvatura especificada por las constantes k y w . La constante k sirve para formar los ángulos $\alpha (P(i - k))$ y $\beta (P(i + k))$, como se

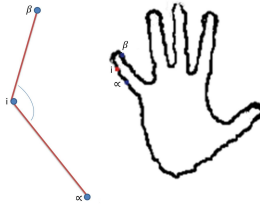


Figura 1. Algoritmo curvatura-k.

puede observar en la Figura 1. La constante w es un umbral medido en radianes. En la aplicación de este artículo los valores de k y w son 20 y 0,95993 radianes, respectivamente. El valor de k se encontró por prueba y error, y el ángulo w al, manualmente, medir los valores de curvatura-k en dedos.

Este método es preciso y rápido pero tiene un gran problema, la definición de las constantes k y w . En especial, encontrar el valor de la constante k es más complicado porque el rango de profundidad (distancia entre la cámara y el usuario) donde un valor específico de k funciona es muy corto.

Aproximación de contorno polígonos En [2] proponen aproximar con polígonos el contorno de las manos, hasta el punto en el que cada dedo se representa con dos vectores. Después a este nuevo contorno aproximado se aplica el algoritmo *ConvexHull*. Los vertices de los polígonos que se encuentran sobre el contorno encontrado por el algoritmo *ConvexHull* son considerados como dedos.

El problema que presenta este algoritmo es similar al anterior, se tiene que calibrar para cada aplicación en especial. Esto se debe a que la aproximación con polígonos debe de ser ajustada dependiendo de la distancia entre la cámara y el usuario. Se realizaron pruebas y no se consiguió obtener un nivel de aproximación que funcionara para el rango de visión de nuestro sensor de profundidad.

3.2. Algoritmos de reconocimiento de objetos

Estos métodos usan algoritmos supervisados de aprendizaje, por lo tanto, se necesita entrenar el sistema antes de poder usarlo. Adaboost [6], es el algoritmo más popular para esta tarea porque es rápido y preciso. Esto se debe a que usa clasificadores débiles para determinar si una imagen contiene o no el objeto de interés. Aunque este método es preciso, es muy difícil conseguir una buena base de datos de clasificación por todo el trabajo que se requiere y algunos problemas a los que puede ser susceptible, como el sobre-entrenamiento. Tampoco es tan rápido como al algoritmo curvatura-k.

3.3. Emparejamiento de modelos

La idea básica del emparejamiento de modelos es crear el modelo de un objeto de interés, en este caso manos y dedos, y luego buscar en la imagen

que se quiera objetos que se emparejen con el modelo. El modelo puede ser creado usando emparejamiento de bordes, emparejamiento en escala de grises o emparejamiento de gradiente. Para atacar el problema de la rotación y variación de escala, este método necesita buscar usando diferentes modelos. Para un rango específico es preciso pero el buscar en una imagen el modelo sin importar la escala o rotación, reduce sustancialmente el desempeño de este algoritmo.

3.4. Cálculos morfológicos

La premisa de este método es que los dedos son largos y delgados en comparación con la mano. Por lo tanto, desaparecen cuando un operador de erosión es aplicado a una imagen que contenga una mano. El operador de abrir puede ser utilizado para detectar dedos al escoger una ventana apropiada y luego detectando la resta con la imagen original. El resultado de la resta son blobs de dedos. Como sucede con los métodos previamente analizados, este algoritmo necesita ser ajustado para que funcione en una aplicación en específico.

4. Nuestro sistema

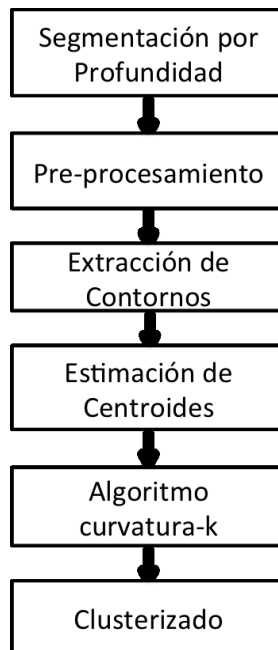
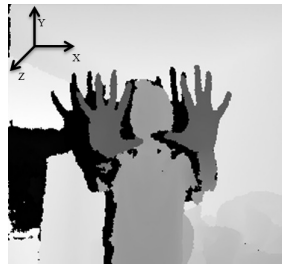
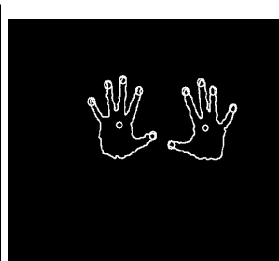
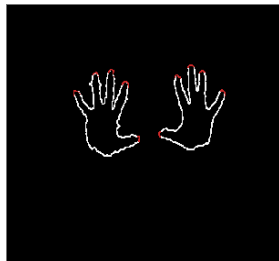


Figura 2. Diagrama de flujo del algoritmo de selección de Dedos.



(a) *Izquierda*: Mapa de Profundidad Original. *Centro*: Segmentación por Profundidad. *Derecha*: Pre-procesamiento



(b) *Izquierda*: Extracción de Contornos. *Centro*: curvatura-k. *Derecha*: Centroides de las Manos y Detección de Dedos

Figura 3. Resultados de cada uno de los pasos de nuestro algoritmo.

La Figura 2 muestra los pasos que nuestro método sigue, también podemos observar la imagen resultante de cada paso. En las siguientes sub-secciones se explicará cada paso en detalle:

1. *Segmentación por Profundidad*: Asumiendo que el objeto más cercano a la cámara es el objeto de interés, manos, usamos un filtro de profundidad. Este filtro elimina de la escena todos los píxeles que están fuera del rango $Rango(PC-l)$, donde PC es el valor, en cm, del píxel más cercano a la cámara. De manera experimental se encontró el valor óptimo de l que funciona en el rango de nuestra cámara de profundidad. Este valor óptimo encontrado fue de $35cm$ y en la Figura 3(a) podemos ver la imagen que obtenemos al aplicar este paso.
2. *Pre-procesamiento*: Primero, un filtro Gaussiano es usado para reducir el ruido de la cámara y asegurarnos de que píxeles aislados no formen parte de nuestro objeto de interés. Luego un filtro de cerrado morfológico es aplicado para llenar huecos y suavizar las orillas externas (Figura 3(a)).
3. *Extracción de Contornos*: Los contornos son extraídos usando la implementación de OpenCV de *Approximate Freeman Chains* [7] (Figura 3(a)). Para los pasos siguientes solamente vamos a tomar en cuenta aquellos contornos

que sean más grandes que un umbral dado. Este umbral se encuentra de forma experimental y funciona en el rango de 0.3 a 3.5 m.

4. *Estimación de Centroides*: Para distintas aplicaciones es útil conocer el centroide de las manos (Figura 3(b)) para seguir su posición. Para obtener su centroide, es fácil el calcular los momentos H_u [8] para cada uno de los contornos con las siguientes fórmulas:

$$\begin{aligned} A &= m_{0,0} \\ x_c &= \frac{m_{1,0}}{A} = \frac{m_{1,0}}{m_{0,0}} \\ y_c &= \frac{m_{0,1}}{A} = \frac{m_{0,1}}{m_{0,0}} \end{aligned}$$

dónde (x_c, y_c) es el centro de gravedad del objeto

5. *Algoritmo curvatura-k*: Por su velocidad y precisión es que seleccionamos este algoritmo para detectar dedos. Su única desventaja es encontrar el valor adecuado de la constante k . Para resolver este problema se definió una constante k general que funciona en el rango de 0,3 a 3,5m que nuestro sensor de profundidad maneja. Usando nuestro valor k general (15) conseguimos entre 10 y 20 puntos de interés por dedo. Podemos ver el resultado en la Figura 3(b).
6. *Clusterizado*: Para conseguir la posición 3D de cada dedo es necesario clusterizar los puntos obtenidos en el paso anterior. Para conseguir nuestro objetivo y mantener la velocidad del algoritmo de detección de dedos, se implementó una versión modificada de vecinos cercanos para realizar esta tarea. La idea básica de este método es usar el algoritmo de vecinos cercanos para centrar los centros que el algoritmo AS compute, este proceso se repite hasta que todos los puntos estén clasificados o se llegue a un máximo de iteraciones. En las siguientes líneas se muestra el pseudocódigo de este algoritmo:

```
// Repetir hasta que se llega al umbral
while continuar
    // Computar el algoritmo de vecinos cercanos
    CentrosInfo = vecCer(centros, inf, umbralK)

    // Computar el algoritmo AS
    [centrosNuevos] = as(centrosInfo, umbralAS)

    // Revisar si se tiene nuevos centros
    if centrosNuevos == centros
        continuar = 0
        break
    end
end
```

5. Resultados

Se le pidió a 4 personas que probaran nuestro sistema y conseguimos un 95,17% de efectividad con un 3,2% de falsos positivos en 200 cuadros que fueron analizados. Es difícil comparar directamente el desempeño de nuestro método con otros porque no hay un estándar para evaluar este tipo de aplicaciones. Pero podemos concluir que:

- Es más robusto que métodos que usan cámaras RGB
- Es más robusto que métodos que necesitan calibrarse para cada aplicación

6. Conclusiones y trabajo a futuro

Nuestro objetivo se cumplió. Este artículo presenta un sistema de detección de manos y dedos que usa mapas de profundidad y análisis de contorno adaptivo. Nuestra técnica puede detectar manos y dedos en todo el rango en el que nuestro sensor funciona. Fue implementado en CUDA la mayor parte de los algoritmos, con lo que conseguimos una aplicación de visión computacional en tiempo real. Tampoco es afectado por cambios de iluminación, reflejos o sombras. Como trabajo a futuro desarrollaremos aplicaciones que hagan uso del sistema propuesto en este artículo. Principalmente nos interesa trabajar en interfaces naturales que detecten gestos de manos y dedos en 3D.

Referencias

1. Trigo, T.R., Pellegrino, S.R.M.: An analysis of features for hand-gesture classification. 17th International Conference on Systems, Signals and Image Processing (2010)
2. L.C. Ebert, G. Hatch, M.T.S.R.: Invisible touch—control of a dicom viewer with finger gestures using the kinect depth camera. *Journal of Forensic Radiology and Imaging*, Volume 1, Issue 1 (2013)
3. Yun, L., Peng, Z.: An automatic hand gesture recognition system based on violajones method and svms. 2009 Second International Workshop on Computer Science and Engineering (2009)
4. H. Koike, Y. Sato, Y.K.H.T., Kobayash, M.: Visual tracking of bare fingers for interactive surface. *ACM Symposium on User interface Software and Technology (UIST '04)* (2004)
5. Dung Duc Nguyen, T.C.P., Jeon, J.W.: Fingertip detection with morphology and geometric calculation. *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2009)
6. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* (1997)
7. Seo, N.: Tutorial: Opencv haartraining (rapid object detection with a cascade of boosted classifiers based on haar-like features). <http://note.sonots.com/SciSoftware/haartraining.html> (2008)
8. Hu, M.K.: Visual pattern recognition by moment invariants. *IRE Trans. Inform. Theory* (1962)

Análisis de componentes principales aplicado al proceso de reconocimiento de objetos 3D bajo rotación

Manuel Ornelas, Valentín Calzada, Hector Puga, Raúl Santiago y Martín Carpio

División de Estudios de Posgrado e Investigación, Instituto Tecnológico de León,
León Guanajuato, México
valecalzada@hotmail.com

Resumen El presente trabajo propone un método para caracterizar y reconocer objetos 3D, basado en la identificación de patrones obtenidos a partir del análisis de la deformación de una línea de luz láser proyectada sobre un objeto en rotación, sin necesidad de una reconstrucción tridimensional. Las imágenes de líneas de luz láser fueron capturadas mediante una cámara CCD, posteriormente se pre-procesaron y segmentaron. Durante el proceso de caracterización se obtuvieron vectores característicos de dimensión \mathbb{R}^{144} , por lo que se realizó un análisis de componentes principales para reducirlos a \mathbb{R}^{65} . Al final, los procesos de reconocimiento de patrones se llevaron a cabo mediante los algoritmos K-Nearest Neighbor, Naïve Bayes y una Red Neuronal Artificial feedforward, en donde se utilizó la validación cruzada para comparar el desempeño de cada uno, utilizando como instancias de prueba la base de datos original y la reducida por el análisis de componentes principales, logrando porcentajes de exactitud superiores al 88 %.

Palabras clave: visión por computadora, reconocimiento de patrones, luz estructurada, análisis de componentes principales, momentos invariantes de Hu, medidas de compacidad.

1. Introducción

En la actualidad, existen una gran cantidad de técnicas para reconstruir o reconocer objetos 3D a partir del análisis de imágenes bidimensionales, técnicas abordadas por un área de investigación llamada visión por computadora. Una de estas técnicas hace uso de los sistemas de luz estructurada, en donde se realizan proyecciones de franjas o de luz láser para analizar y caracterizar las superficies de los objetos bajo estudio [1-12].

En este artículo se propone un método para caracterizar y reconocer objetos tridimensionales rotados, a partir del análisis de la deformación de una línea de luz láser proyectada sobre la superficie de los mismos. Por medio de una cámara CCD se realiza la captura de imágenes de una línea vertical de luz láser proyectada en distintas posiciones del objeto en rotación. Cada imagen es pre-procesada y segmentada mediante técnicas de procesamiento de imágenes. Para la caracterización se utilizan descriptores de forma, como las medidas de compacidad y los momentos invariantes de Hu, con los cuales se definen los vectores característicos que describen el objeto. Estos vectores resultantes son de una dimensión considerable, por lo que se realiza un análisis de componentes principales para su reducción, con la finalidad de disminuir el coste

computacional que implica el procesamiento de los mismos. Para el proceso de reconocimiento se utilizan clasificadores supervisados, tales como K-Nearest Neighbor y Naïve Bayes, así como el uso de una Red Neuronal Artificial feedforward con su entrenamiento por backpropagation. Al final, se compara el desempeño de los algoritmos mediante el uso de la validación cruzada, utilizando como instancias de prueba la base de datos con los vectores característicos originales y la base de datos con los vectores reducidos por medio del análisis de componentes principales.

El presente artículo se encuentra organizado de la siguiente manera: En la segunda sección se exponen los conceptos teóricos y las herramientas utilizadas durante el desarrollo de la investigación, en la tercera sección se muestra una descripción general del método empleado, en la cuarta sección se muestra el proceso y desarrollo utilizado, en la quinta sección los resultados obtenidos y para finalizar, en la sexta sección las conclusiones.

2. Conceptos teóricos

La visión por computadora es un área de investigación en la cual, uno de sus objetivos, es modelar computacionalmente los procesos de percepción visual en los seres vivos, de manera que permitan simular estas capacidades visuales a través de una computadora mediante el uso de imágenes digitales. Dentro del campo de la visión por computadora existen diversas técnicas que son empleadas con éxito en numerosas aplicaciones, entre ellas se encuentran los sistemas de luz estructurada, cuya geometría básica es una triangulación.

A continuación se exponen algunos conceptos teóricos usados en este trabajo.

2.1. Segmentación

La segmentación de una imagen consiste en simplificar y/o cambiar su representación en otra más significativa y más fácil de analizar. El método de segmentación utilizado en esta investigación es el de Otsu, que se fundamenta en la técnica del análisis discriminante al maximizar alguna medida que permita separar clases: la de los objetos y la del fondo [13].

2.2. Caracterización

El objetivo de la caracterización es encontrar una representación cuantitativa de una imagen, que sea analizable a través de una computadora [14]. Los descriptores utilizados para caracterizar las imágenes en este trabajo son los momentos invariantes de Hu y las medidas de compacidad, los cuales se describen a continuación.

Momentos invariantes de Hu: Hu definió 7 momentos capaces de medir la dispersión de píxeles en una imagen digital. Estos momentos se mantienen invariantes ante rotaciones, traslaciones y cambios de escalas en los objetos representados por regiones binarias en una imagen [15].

Compacidades: La compacidad es un parámetro cuantitativo de la forma y se calcula mediante la interacción entre el área (n) y perímetro (P) de una imagen digital [16]. Las medidas de compacidad empleadas son:

1. **Compacidad clásica:**

$$C_c = \frac{\text{perimetro}^2}{\text{area}} . \quad (1)$$

2. **Compacidad discreta normalizada:**

$$C_{DN} = \frac{C_D - C_{D_{min}}}{C_{D_{max}} - C_{D_{min}}} . \quad (2)$$

donde

$$C_D = \frac{4n-P}{2}; C_{D_{min}} = n - 1; C_{D_{max}} = \frac{4n-4\sqrt{n}}{2} .$$

3. **Factor E:**

$$Factor_E = \frac{2P}{4n - P} . \quad (3)$$

4. **Factor E normalizado:**

$$Factor_{EN} = \frac{P}{4\sqrt{n}} . \quad (4)$$

5. **Peura:**

$$Peura = \frac{P_{circulo}}{P_{forma}} . \quad (5)$$

Peura relaciona la silueta del objeto digital y lo compara con un círculo. Este círculo está determinado por el área de la silueta del objeto [17].

2.3. Enfoques de reconocimiento de patrones

El reconocimiento de patrones es la asignación de una etiqueta a un determinado valor de entrada. Un ejemplo de reconocimiento de patrones es la clasificación, que pretende asignar cada valor de entrada a uno de un conjunto dado de clases. Para realizar los procesos de clasificación, existen diferentes algoritmos clasificadores supervisados, como el K-Nearest Neighbor y el Naïve Bayes. Otro enfoque de reconocimiento de patrones son las Redes Neuronales Artificiales feedforward con su entrenamiento por backpropagation, las cuales tratan de emular el comportamiento biológico del cerebro, a través de una interconexión de neuronas artificiales que simulan el proceso de sinapsis.

K-Nearest Neighbor: Consideremos m clases $\{C_i\}_{i=1}^m$ y un conjunto N de patrones de muestra $\{y_i\}_{i=1}^N$, cuya clasificación es conocida a priori. Sea x un patrón arbitrario entrante, el enfoque de Nearest Neighbor (vecino mas cercano) asigna a x a la clase donde el patrón se encuentra más cerca del conjunto $\{y_i\}_{i=1}^N$, es decir, si

$$\|x - y_j\| = \min_{1 \leq i \leq N} \|x - y_i\| . \quad (6)$$

entonces $x \in C_j$. Este esquema es básicamente otro tipo de clasificación por mínima distancia, se puede modificar mediante la consideración de los k vecinos más cercanos a x y el uso de un clasificador tipo regla-mayoritaria [18].

Naïve Bayes: Consideremos un número finito de clases $\{C_1, C_2, \dots, C_m\}$ y un vector característico x en R^n . Cada componente de este vector es una característica escalar significativa de C_i , $1 \leq i \leq m$. El vector característico es una variable aleatoria dada una distribución de probabilidad condicional $p(x|C_i)$, $1 \leq i \leq m$. Si denotamos a $p(x|C_i)$ como la probabilidad a priori que tiene un patrón entrante de pertenecer a una clase C_i , entonces la probabilidad a posteriori de este patrón con un vector característico adjunto x para pertenecer a C_i , está dada por la siguiente fórmula de Bayes:

$$p(C_i|x) = \frac{p(C_i)p(x|C_i)}{p(x)}, 1 \leq i \leq m . \quad (7)$$

donde

$$p(x) = \sum_{i=1}^m p(C_i)p(x|C_i) . \quad (8)$$

es la probabilidad de distribución de x [18].

Red Neuronal Artificial- Backpropagation: El método backpropagation es un algoritmo de entrenamiento de redes neuronales artificiales que aplica el método de descenso de gradiente para minimizar un error producido por una red neuronal. El entrenamiento de la red consiste en una retropropagación que consta de las siguientes etapas:

1. Alimentar la entrada con un patrón de entrenamiento en la red neural.
2. Análisis del error en la retropropagación.
3. Actualización de los pesos.

Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total [19].

2.4. Análisis de Componentes Principales

El análisis de componentes principales (ACP) es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos. Intuitivamente la técnica sirve para hallar las causas de la variabilidad de un conjunto de datos y ordenarlas por importancia.

Formalmente, supongamos que existe una muestra con n individuos para cada uno de los cuales se han medido m variables (aleatorias). El ACP permite encontrar un número de factores subyacentes $p < m$ que explican aproximadamente el valor de las m variables para cada individuo. El hecho de que existan estos p factores subyacentes puede interpretarse como una reducción de la dimensionalidad de los datos, donde antes necesitábamos m valores para caracterizar a cada individuo ahora nos bastan p valores. Cada uno de los p valores encontrados se llama componente principal, de ahí el nombre del método.

Existen dos formas básicas para aplicar el ACP, la primera es utilizando la matriz de correlación cuando los datos no son dimensionalmente homogéneos o el orden de magnitud de las variables aleatorias medidas no es el mismo. La segunda, es utilizando la matriz de covarianzas cuando los datos son dimensionalmente homogéneos y presentan valores medios similares [22].

3. Descripción general del método

Sabemos por los procesos de reconstrucción tridimensional [20], que existe información en las líneas de luz láser proyectadas sobre superficies de objetos; este trabajo propone un método de análisis de la deformación de esas líneas de luz, para realizar reconocimiento de objetos, sin necesidad de una reconstrucción tridimensional. En primera instancia, se obtiene una imagen de la proyección de la línea de luz láser por medio de una cámara CCD, obteniendo una representación bidimensional de una pequeña porción del objeto en una posición inicial. Para obtener la representación total del objeto, se hace un barrido o escaneo mediante la rotación del mismo, en donde se realiza una captura de la línea vertical de luz láser cada 30° . Así, por cada objeto se obtienen un conjunto de 12 líneas que corresponden a los 360° que conforman la rotación completa del mismo. Una vez obtenido el conjunto de imágenes por objeto, se realiza el pre-procesamiento y la segmentación de cada una de éstas. Posteriormente se hace la caracterización de cada una de imágenes de líneas de luz y se genera un vector característico por objeto. Los descriptores utilizados para generar el vector característico (patrón) son los siete momentos invariantes de Hu y cinco medidas de compacidad, las cuales son: Compacidad clásica, Compacidad discreta normalizada, Factor E, Factor E Normalizado y Peura. Una vez obtenido el conjunto de vectores característicos de todos los objetos, se realiza un análisis de componentes principales para reducir la dimensionalidad de los mismos. Al final, se analiza el conjunto de vectores mediante el uso de diferentes enfoques de reconocimiento de patrones, como clasificadores supervisados y redes neuronales artificiales.

4. Desarrollo

Las ocho clases de objetos utilizadas para realizar el proceso de reconocimiento son: botellas, figurillas (porcelana), huevos, manzanas, piedras, plátanos, tazas y vasos. Cada clase está conformada por 15 objetos del mismo tipo, pero de diferentes tamaños y forma, obteniendo una población total de 120 objetos a analizar.

4.1. Captura de imágenes

Este proceso consiste en la captura, mediante una cámara CCD, de las imágenes de líneas de luz láser proyectadas sobre los objetos de estudio. Para ello se realiza un sistema de triangulación [21] en los que interactúan objeto, láser y cámara (Figura 1).

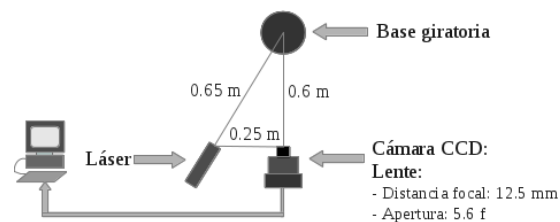


Figura 1. Diagrama del sistema de adquisición de imágenes

Durante este proceso, se realiza la proyección de una línea vertical de luz láser sobre la superficie del objeto colocado en la base giratoria. Dada la proyección de la luz, se hace un barrido del objeto mediante la rotación de los 360° de la base, en donde cada 30° se realiza una captura de la línea de luz láser a través de la cámara CCD, obteniendo un conjunto de 12 imágenes por objeto en cada barrido. En total se realizó el mismo barrido a 120 objetos, dando un total de 1440 imágenes para procesar.

En la Figura 2, se muestran imágenes capturadas de líneas de luz proyectadas sobre objetos de diferentes clases.

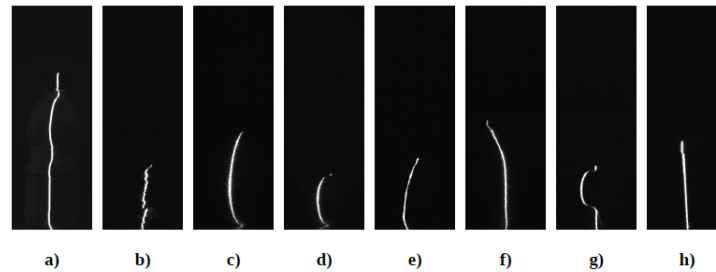


Figura 2. Imágenes representativas de la línea de luz proyectada sobre objetos de cada clase: a) Botella, b) Figurilla, c) Huevo, d) Manzana, e) Piedra, f) Plátano, g) Taza, h) Vaso

4.2. Pre-procesamiento de imágenes

El siguiente paso consiste en aplicar un suavizado gaussiano a cada imagen, con la finalidad de eliminar pixeles ruidosos que puedan alterar la continuidad de la línea de luz capturada. Ya que las líneas de luz láser son de color blanco en las imágenes (Figura 2), por preferencia se invierte la imagen para facilitar el procesamiento de los algoritmos de caracterización implementados. Este proceso se muestra en la Figura 3.

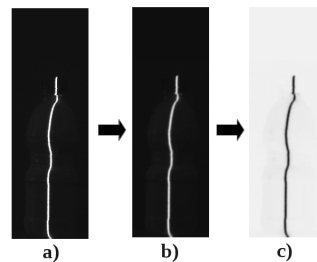


Figura 3. Proceso de pretratamiento de imágenes: a) Imagen Original, b) Imagen tratada con un filtro Gaussiano, c) Imagen invertida o negativo

4.3. Proceso de segmentación

La segmentación se realiza mediante la binarización por el método de Otsu, que a pesar de ser un algoritmo básico, realiza una correcta segmentación de las imágenes. En la Figura 4, se muestra el resultado de la segmentación de una línea de luz.



Figura 4. Línea segmentada mediante el método de Otsu

Para el conjunto de imágenes tratadas, los umbrales calculados por Otsu oscilan entre 160 y 170, en un rango de 0 a 255 niveles de gris.

4.4. Proceso de caracterización

Este proceso consiste en el cálculo de los descriptores que representarán las líneas de luz de una manera vectorial (patrón) para el posterior análisis mediante técnicas de Reconocimiento de Patrones. El vector para una línea, está conformado por 12 descriptores d_n , los cuales son los siete momentos invariantes de Hu y cinco medidas de compacidad. Así, para representar la primera línea (l_1) perteneciente a la imagen en la posición inicial (0°) de cada objeto, se tiene un vector similar al siguiente:

$$\vec{l}_1 = [d_1, d_2, d_3, \dots, d_{12}] . \quad (9)$$

El vector que representa la línea 1 (\vec{l}_1) es de dimensión \mathbb{R}^{12} . Para obtener el vector característico (Vc) de todo el objeto, se concatenan los vectores de las 12 líneas:

$$\vec{Vc} = \vec{l}_1 + \vec{l}_2 + \vec{l}_3 + \dots + \vec{l}_{12} . \quad (10)$$

En total se tienen un conjunto de 120 vectores característicos (15 por clase), cada uno de dimensión \mathbb{R}^{144} , y todos se almacenan en una base de datos, en donde a cada vector se le asigna una etiqueta de la clase a la que pertenece.

4.5. Proceso de ACP

Debido a que la dimensionalidad del vector característico por objeto es de \mathbb{R}^{144} , se realizó un análisis de componentes principales a toda la base de datos, con la finalidad de reducir su dimensionalidad [22]. El primer paso es obtener la matriz de correlación R de la base de datos a analizar, ya que el orden de magnitud de los descriptores utilizados

para caracterizar los objetos no es el mismo. Posteriormente se realiza la descomposición espectral de la matriz R para obtener sus eigen-valores y sus eigen-vectores. A partir de éstos se obtienen los componentes principales, de los cuales se seleccionan aquellos cuyo porcentaje de variabilidad explicado acumulado es igual o superior al 90 %.

Después, para obtener las variables representativas de los componentes elegidos, se aplica un criterio de selección a partir de un porcentaje acumulado, éste consiste en ordenar de mayor a menor los coeficientes de cada variable de la componente principal seleccionada, se calcula el porcentaje individual de cada variable en función de su coeficiente, en donde éste representa la variabilidad.

Al final, se realiza la suma de todos los porcentajes individuales hasta llegar a un valor acumulado igual o mayor a una cota establecida. Para estas pruebas se estableció una cota empírica del 90 %, el cual es un porcentaje adecuado que permitió una reducción significativa de los componentes de los vectores característicos.

Para el análisis realizado en este trabajo, se requirieron 65 variables para alcanzar la cota establecida del 90 %, por lo que éstas se seleccionan para formar el nuevo vector característico de dimensión \mathbb{R}^{65} .

4.6. Reconocimiento

Una vez almacenados todos los vectores en una base de datos con sus respectivas etiquetas por clases, se prosigue a realizar el proceso de reconocimiento mediante el uso de diferentes algoritmos como K-Nearest Neighbor (KNN), Naïve Bayes y la Red Neuronal Artificial (RNA) FeedForward con su entrenamiento por backpropagation, así como el uso de la validación cruzada para comparar el desempeño de cada algoritmo [23], en donde se realizaron 10 particiones (K -folds) para la selección de las muestras para su entrenamiento y recuperación.

Las pruebas de reconocimiento se realizaron para dos bases de datos, la primera de dimensión \mathbb{R}^{144} y la segunda, su versión reducida por el ACP, de dimensión \mathbb{R}^{65} .

5. Resultados

A continuación se muestran los resultados obtenidos para las dos bases de datos. Como se mencionó en la sección anterior, se utilizó la validación cruzada para seleccionar las muestras de entrenamiento y recuperación para los algoritmos implementados, donde se utilizó un valor incremental de K -folds empezando con 2 y terminando con 10. Después de realizar diversas pruebas en ambas bases de datos con los algoritmos de reconocimiento de patrones, se establecieron los mejores parámetros: en la base de datos original, para KNN $K=1$, para la RNA FeedForward $\alpha=0.6$, 1 capa con 144 entradas, 1 capa oculta con 144 neuronas y 1 capa de salida con 8 neuronas. En la base de datos reducida por ACP, para KNN $K=1$, para la RNA FeedForward $\alpha=0.6$, 1 capa con 65 entradas, 1 capa oculta con 65 neuronas y 1 capa de salida con 8 neuronas. En la Tabla 1, se muestran los resultados de exactitud para cada K -fold de la validación cruzada en la base de datos original, así como la exactitud promedio.

Tabla 1. Porcentajes de exactitud para la base de datos original.

K-Fold	KNN	Naïve Bayes	RNA
2	74.5 %	89.28 %	85.05 %
3	82.33 %	95.0 %	84.66 %
4	88.5 %	96.87 %	89.79 %
5	87.4 %	95.0 %	87.98 %
6	95.83 %	94.79 %	87.0 %
7	94.57 %	97.32 %	94.52 %
8	93.75 %	93.75 %	90.99 %
9	94.44 %	95.83 %	91.99 %
10	97.5 %	95.0 %	91.95 %
Exactitud promedio	89.86 %	94.76 %	89.32 %

En la Tabla 2, se muestran los resultados de exactitud obtenidos para la segunda base de datos reducida por ACP y su exactitud promedio.

Tabla 2. Porcentajes de exactitud para la base de datos reducida por ACP.

K-Fold	KNN	Naïve Bayes	RNA
2	77.5 %	84.82 %	84.53 %
3	78.66 %	90.0 %	91.89 %
4	91.5 %	91.66 %	88.08 %
5	86.6 %	93.33 %	88.09 %
6	95.83 %	94.79 %	91.93 %
7	95.42 %	96.25 %	87.22 %
8	93.75 %	92.18 %	92.12 %
9	93.0 %	94.44 %	94.52 %
10	97.5 %	96.42 %	95 %
Exactitud promedio	89.97 %	92.65 %	88.66 %

6. Conclusiones

En este trabajo se muestra que es posible realizar reconocimiento de objetos 3D bajo rotación a través del análisis de la deformación de las líneas de luz, sin necesidad de realizar una reconstrucción tridimensional. El ACP redujo significativamente la dimensionalidad del vector característico de un espacio en \mathbb{R}^{144} a \mathbb{R}^{65} , facilitando su procesamiento computacional. Los promedios del porcentaje de exactitud en ambas bases de datos fueron similares, asegurando que el vector característico reducido puede describir correctamente el objeto. Para esta serie de pruebas de reconocimiento de patrones, todos los algoritmos implementados obtuvieron un porcentaje de exactitud promedio superior al 88 %, de los cuales, el clasificador Naïve Bayes fue el de mejor desempeño, obteniendo una exactitud promedio de 94.76 % en la base de datos original

y 92.65 % en la base de datos reducida por el ACP.

Agradecimientos. Los autores agradecen a CONACYT y DGEST por los apoyos proporcionados para la realización de esta investigación.

Referencias

1. R. Gonzalez, R. Woods: Digital image processing, 2nd edition, Prentice Hall, pages 643-750, 2002.
2. R. Pinto, H. Sossa: Automatic 2-D and 3-D object localization on images, *Computación y sistemas*, Número especial, pages 26-34, CIC-IPN. ISSN 1405-5546, 2002.
3. A. Patiño, D. Miranda, J. Meneses: Scanner 3D de objetos a 360° de observación, *revista colombiana de física*, Vol. 35, No.1. 2003.
4. B. Javidi and E. Tajahuerce: Three-dimensional object recognition by use of digital holography, *Opt. Lett.* 25, 610-612 (2000)
5. Y. Wanga, Y. Lia, J. Zhoua, J. Zhanga, J. Fanga: A non-encoding structured light approach with infrared illumination for 3D large field shape measurement, *Optics and Laser Technology*, Vol. 49, pages 28-32, 2013.
6. F. Samadzadegana, A. Azizia, M. Hahn, C. Lucasa: Automatic 3D object recognition and reconstruction based on neuro-fuzzy modelling, *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 59, Issue 5, Pages 255–277, 2005
7. M. Brown: Unsupervised 3D object recognition and reconstruction in unordered datasets, *3-D Digital Imaging and Modeling 2005, Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, pages 56-63
8. F. Rothganger: 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints, *Computer Vision and Pattern Recognition*, Vol. 2, pages II- 272-7, 2003.
9. J. Salvi, S. Fernandez, T. Pribanic, X. Lladó: A state of the art in structured light patterns for surface profilometry, *Pattern Recognition* 43(8): 2666-2680, 2010.
10. A. E. Johnson, M. Hebert: Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* 21(5): 433-449, 1999.
11. A. Frome, D. Huber, R. Kolluri, T. Bülow, J. Malik: Recognizing Objects in Range Data Using Regional Point Descriptors, *ECCV*, 2004.
12. J. Tangelder, R. C. Veltkamp: A Survey of Content Based 3D Shape Retrieval Methods, *Proceedings of the Shape Modeling International 2004 (SMI'04)*.
13. N. Otsu: A threshold selection method from gray-level histograms. *Automatica*, 11:285–296, 1975.
14. H. Sossa: Rasgos descriptores para el reconocimiento de objetos, *Ciencia de la computación*, CIC IPN, pages 10–29, México 2002.
15. M.K. Hu: Visual pattern recognition by moment invariants, *IRE Transactions on information theory*, 8(2) pages 179–187, 1962.
16. E. Bibriesca: Measuring 2-D Shape Compactness Using the contact perimeter, *Computer Math Applic.* Vol. 33, No. 11, pages 1-9, 1997.
17. R. Santiago: Descriptores de Regiones Binarias en Z2 y Z3, Basados en el Concepto de Compacidad, IIMAS, Universidad Nacional Autónoma de México, Tesis doctoral, México, 2008.
18. M. Friedman, A. Kandel: Introduction to pattern recognition, World Scientific, Volume: 32, pages 65–106, 1999.
19. R. Rojas: Neural Networks: A Systematic Introduction, Springer, pages 151-155, 1996.

20. J. Espinal, M. Ornelas: Comparación de diferentes métodos usados en la obtención de la posición de una línea de luz para reconstrucción 3D, Memoria del VI taller-escuela de procesamiento de imágenes PI 2009, México.
21. Daniel V. Hahn: Non-laser-based scanner for three-dimensional digitization of historical artifacts, Applied Optics, Volume 46, No. 15, 2007.
22. Jolliffe I.T: Principal Component Analysis, Springer Series in Statistics, 2nd ed., XXIX, page 487, ISBN 978-0-387-95442-4, NY, 2002.
23. Ron Kohavi: A study of cross-validation and Bootstrap for accuracy estimation and model selection, International joint conference on artificial intelligence (IJCAI), 1995.

Algoritmo de segmentación para imágenes reales para la clasificación de cetáceos

D. B. Trejo-Salazar^{1,2}, B. E. Carvajal-Gómez^{1,2} y F. J. Gallegos-Funes³

¹Escuela Superior de Cómputo, Instituto Politécnico Nacional,
Juan de Dios Bátiz s/n Professional U. Adolfo López Mateos, SEPI-ESCOM
Gustavo A. Madero, 07738, México DF

²Unidad Profesional en Ingeniería y Tecnologías Avanzadas, Instituto Politécnico Nacional,
Av. IPN no. 2580 Col. Barrio La Laguna Ticoman, 07340, México DF

³Escuela Superior de Ingeniería Mecánica y Eléctrica, Instituto Politécnico Nacional,
Av. IPN s/n, ESIME SEPI-Electrónica, Lindavista, 07738, México DF

{dtrejo, becarvajal}@ipn.mx

Resumen La foto-identificación de los cetáceos se puede realizar mediante la observación de sus patrones de pigmentación y forma de la aleta caudal estas se realizan mediante fotografías, estos patrones se pueden ser observados a lo largo de toda la porción del lomo de la ballena. En esta investigación se presenta un algoritmo de segmentación de imágenes para la ayuda de la foto identificación de imágenes reales de la ballena azul en el mar de cortés, aplicando la técnica de la desviación media como umbral de selección de bordes y así como la transformada wavelet discreta (DWT) para la preservación de los detalles finos de la imagen a segmentar. Los resultados obtenidos se presentan en porcentaje de especificidad, sensibilidad y precisión.

Palabras clave: foto-identificación, transformada wavelet discreta, desviación media.

1. Introducción

Los mamíferos son un componente importante en los ecosistemas terrestre y marítimo, principalmente para los ecosistemas marítimos los cetáceos son de suma importancia, pero el conocimiento que se tiene acerca de estos es limitado [4, 8, 16]. Los océanos cubren el 70 % de la superficie de la Tierra, o dicho de otra manera, una superficie que suma el doble de la superficie de Marte o de la Luna. La identificación individual de las ballenas azules está basada en los patrones únicos de pigmentación y de la forma de su aleta dorsal. La identificación de estos individuos se puede realizar mediante la observación de sus patrones de pigmentación y forma de la aleta dorsal estas se realizan mediante la foto identificación, estos patrones se pueden ser observados a lo largo de toda la porción del lomo de la ballena.

La técnica de la foto identificación, aplicado a la ballena azul, ha mostrado ser útil para investigar los patrones de movimiento, y la estimación de la abundancia de la

población y las tasas de supervivencia a través de modelos de marca-recaptura. Una metodología popular para la extracción de características en los métodos de reconocimiento de patrones se puede dividir en dos categorías: métodos supervisados y métodos de supervisión. Los métodos supervisados utilizan un plano de tierra utilizado de referencia para la clasificación de los datos en base a las características dadas [6]. Estos métodos incluyen redes neuronales [7], la red neuronal de propagación hacia atrás [7], el análisis de componentes principales [7], clasificadores k-vecinos más cercanos [7], clasificación de crestas basada en primitivas [7], wavelet Gabor con clasificación supervisada mezclada con una función Gaussiana [7], los operadores de línea y clasificación de vector de soporte [7]. Los métodos no supervisados trabajan sin ningún conocimiento previo de etiquetado. Algunos de los métodos reportados son el algoritmo agrupación fuzzy C-means [7], algoritmo de agrupamiento en base radio (RACAL) [7], estimación de la máxima verosimilitud [7] y el umbral de entropía local con niveles de gris utilizando la matriz de co-ocurrencia [7].

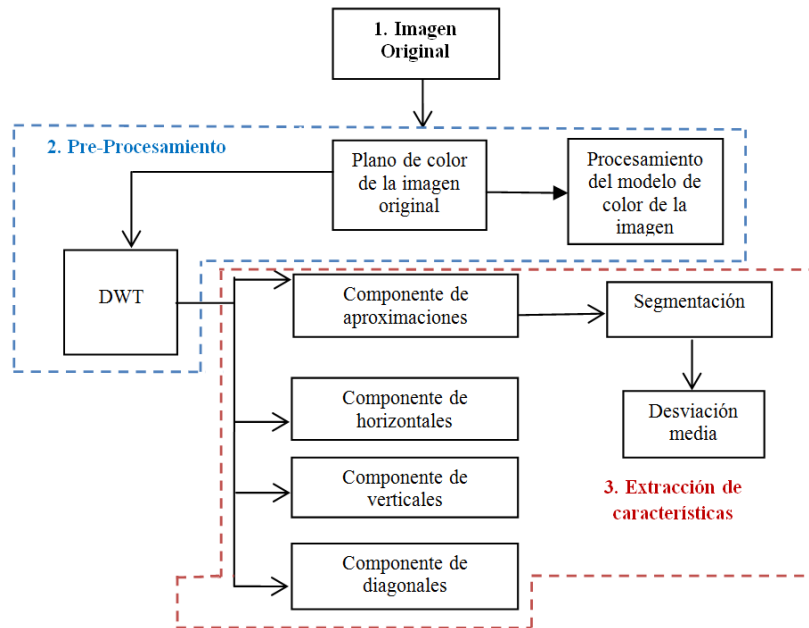


Fig. 1. Diagrama a bloques general del método propuesto.

La metodología de filtrado adaptado explota la aproximación lineal por tramos, la disminución en el diámetro del vaso a lo largo de la longitud vascular y la Gaussiana como perfil de intensidad utiliza un núcleo basado en una función Gaussiana o de sus derivados para mejorar las características de la imagen. El procesamiento morfológico identifica formas específicas, tiene la ventaja de la velocidad y la resistencia al ruido [6, 9, 11, 18]. Los operadores de morfología matemática [6, 12] y filtros combinados [3, 6, 14, 19, 20] son dos ejemplos de esta categoría. En la metodología de clasificación morfológica, destacan estructuras de la línea, como el operador de top-

hat [6, 9, 12]. El uso de la transformada wavelet es otra manera de mejorar las imágenes [10]. Dado que no existe un programa para la ayuda de la segmentación de las ballenas para la ayuda de la identificación y clasificación, esta se lleva a cabo de manera manual, lo cual puede resultar ser muy laborioso cuando el número de individuos es muy grande. El método propuesto en este trabajo es presentado en el diagrama a bloques general, Fig. 1. Se aplica un pre-procesamiento aplicando la transformada wavelet discreta (DWT), y un procesamiento final con la desviación media de cada una de las capas Roja, Verde, Azul (RGB) que integran la imagen, esto ayuda a la extracción de las características del cetáceo principalmente la aleta caudal cuidando evitar así la pérdida de información importante para la identificación y clasificación de este.

El artículo está organizado como sigue: primero se presentó una introducción, en la sección 2 se describe el marco teórico, para el método propuesto, comenzando con una descripción de los rasgos importantes de la foto identificación de la ballena azul para su segmentación, siguiendo con información del formato de las imágenes con las que va a trabajar, continuando con la descripción matemática del método propuesto; en el punto 2.2 se describe de cómo se evaluará el algoritmo planteado, para corroborar si cumple con lo esperado. Continuando con la sección 2.3 donde se muestran los resultados obtenidos y finalizando con una valoración del trabajo propuesto.

2. Método propuesto

Las fotografías se tomaron con una cámara de 35 mm réflex EOS Canon con unos lentes de teleobjetivo de 70-300, con una película en blanco y negro. Dentro de la foto identificación de las ballenas azules, la aleta dorsal es de gran ayuda debido a que esta se encuentra fuera del agua la mayor parte del tiempo. Existen diferentes tipos de aleta dorsal, como se podrá ver en la Figura 2:

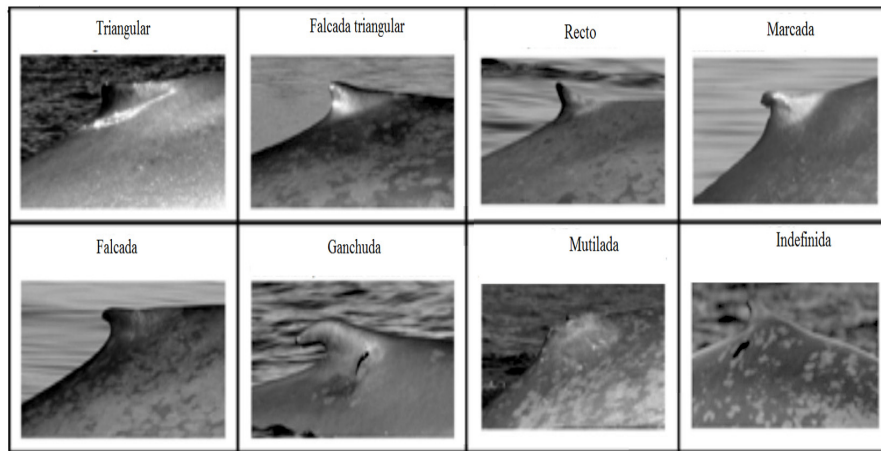


Fig. 2. Diferentes tipos de aleta dorsal.

2.1. Formación y tipos de imágenes

Una imagen puede considerarse como un objeto plano cuya intensidad luminosa y color puede variar de un punto a otro. Se pueden representar como una función continua $f(x, y)$ donde (x, y) son sus coordenadas y el valor de f es proporcional a la intensidad luminosa (nivel de gris) en ese punto Figura 3 [5].

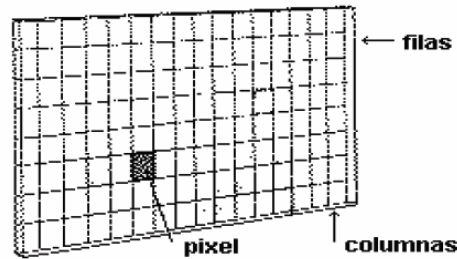


Fig. 3. Representación de una imagen.

La salida de este proceso pueden ser, o bien un único valor (escala de grises) o bien un vector con tres valores por polígono (RGB) que se corresponden con la intensidad de color rojo (R), verde (G) y azul (B) [5]. Una imagen puede considerarse como una matriz cuyos índices de fila y columna identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris de ese punto. Los elementos de una distribución digital de este tipo se denominan elementos de la imagen o más comúnmente píxeles [5]. El método propuesto trabaja en el dominio wavelet y provee un análisis de los coeficientes wavelet en cada banda realizando la estimación del valor de la diferencia de cada pixel de las tres sub matrices de la descomposición de la imagen RGB convertida a escala de grises, identificando de esta manera las variaciones de intensidades de gris distinguiendo de esta manera las áreas donde los píxeles son considerados como ruido (bordes) [1, 2, 10, 17]. Ya que los coeficientes wavelet con gran energía corresponden a la variación abrupta como lo son bordes y texturas, la energía baja corresponde a las regiones lisas, durante el proceso de segmentación se compara el actual núcleo wavelet para encontrar un umbral para identificar las regiones con ruido dominante dentro de la escala. En el método propuesto aquí de segmentación, se propone el siguiente criterio siendo la imagen a segmentar la imagen adquirida por los investigadores del CICIMAR. La diferencia de valores entre cada pixel es elegido como umbral, aplicando así el siguiente criterio: si el valor del pixel del actual núcleo de los coeficientes wavelet es menor al umbral, entonces la respectiva área de la imagen anfitriona es considerada una zona con un valor menor que el de una zona que contiene altos relieves, entonces esta área colocando un 1, sobre esta imagen. De otra manera, los píxeles de esta área son considerados libres de ruido siendo este un valor 0. El método de segmentación es aplicado a cada canal de la imagen a segmentar (R, G, B), este algoritmo es mostrado en la Figura 4. Para el análisis de los coeficientes wavelet, se aplica una descomposición wavelet Haar. De los coeficientes wavelet de las aproximaciones (A) de la imagen a segmentar se calcula la desviación media usando un núcleo de 3×3 ,

Figura 5. La desviación media de los coeficientes en cada banda representa el nivel promedio de la señal en esta escala, y está dada por σ_{ij} ,

$$\sigma_{ij} = \frac{1}{n * m} \left\| \sum_{i=1}^n \sum_{j=1}^m x_{i,j} - \bar{x} \right\| \quad (1)$$

donde x_k es un elemento ij del actual núcleo, $n * m$ es el número de elementos de la muestra, y $\bar{x} = \frac{1}{N} \sum_{k=1}^N x_k$ es el valor medio del actual núcleo .

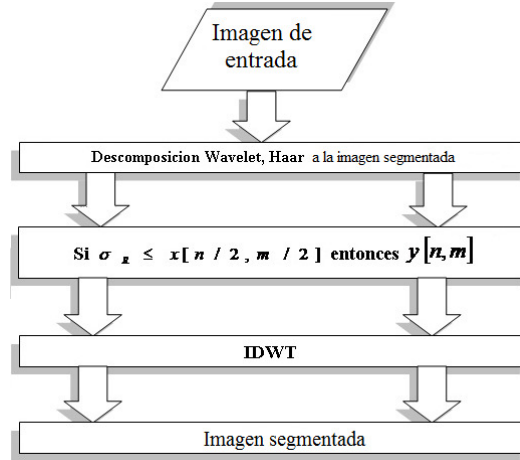


Fig. 4. Algoritmo de segmentación propuesto.

Se aplica la condición $\alpha_s \geq T$ en cada núcleo de las desviaciones estándar $\lambda_s = \sigma_{ij}$, si esta condición se satisface entonces esta área o región es considerada ruidosa y así en el núcleo respectivo de coeficientes wavelet CI_{sij} de la imagen anfitriona puede ser insertada la información a ocultar [1],

$$SI_{sij} = \begin{cases} HI_{ij} & \alpha_s \geq T \\ CI_{sij} & \text{de otra manera} \end{cases} \quad (2)$$

donde HI_{ij} y SI_{sij} son 1 y 0, respectivamente, y α_s es un tipo de ocultamiento propuesto en el método esteganográfico [7, 8],

$$\alpha_s = \text{MED} (\sigma_{ij}) \quad (3)$$

Aquí, $MED(\sigma_{ij})$ denota la mediana de las desviaciones medias del núcleo σ_{ij} . El uso del estimador de mediana en la ecuación (3) representa el filtro mediano que tiene buenas propiedades para la supresión del ruido. De la Figura 4, el bloque de la redundancia en aproximaciones usa un nivel de descomposición con la wavelet Haar [1, 13, 15]. Este algoritmo desvanece las bajas frecuencias de la imagen a segmentar a través de una doble operación de convolución (primero de descomposición, y después la reconstrucción) de los coeficientes y muestras de la imagen, proveyendo más precisión al algoritmo de segmentación preservación de los detalles finos del método de segmentación.

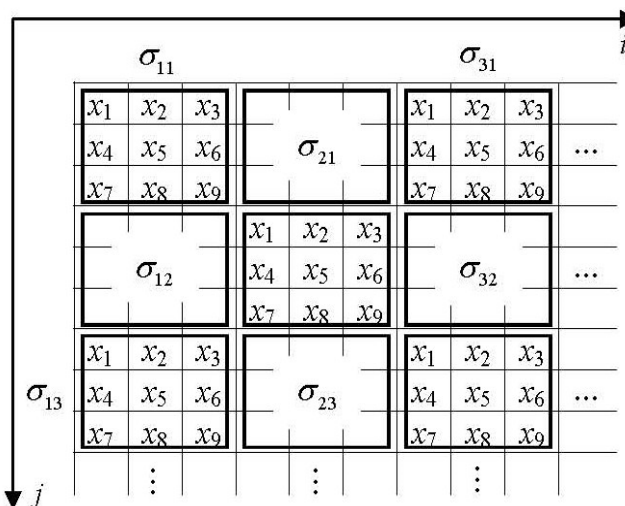


Fig. 5. Procedimiento para calcular la desviación estándar en cada núcleo de los coeficientes ij wavelet de la imagen anfitriona RGB.

2.2. Comparación de resultados obtenidos

El rendimiento del método propuesto se probó con imágenes reales tomadas en su hábitat natural de los cetáceos. Para ver el rendimiento del método propuesto en el desarrollo de las pruebas, se consideraron cuatro casos: dos clasificaciones y dos clasificaciones erróneas. Las clasificaciones son: el verdadero positivo (VP) en la que indica que el algoritmo de identificación decidió correctamente con el cetáceo (es el mismo), falso positivo (FP) en la que indica que el algoritmo de identificación no decidió correctamente (que son diferentes), verdadero negativo (VN) donde indica que el algoritmo de detección decidió correctamente que no es el mismo (si se trata de diferentes cetáceos), falsos negativos (FN) en la que indica que el algoritmo de detección decidió correctamente que se trata de diferentes cetáceos (que son la misma persona). Mediante el uso de estos indicadores se pueden obtener medidas de rendimiento como la sensibilidad, especificidad, valores predictivos y exactitud de la ecuación (6), la ecuación (7) y la ecuación (8) [6, 7]:

$$TRP = TP / \text{conteo de píxeles} \quad (4)$$

$$FPR = FP / \text{no conteo de píxeles} \quad (5)$$

$$\text{Especificidad} = TN / (TN + FP) \quad (6)$$

$$\text{Sensibilidad} = TP / (TP + FN) \quad (7)$$

$$\text{Precisión} = (TP + TN) / \text{píxeles totales} \quad (8)$$

$$PPV = TP / (TP + FP) \quad (9)$$

La tasa verdaderos positivos (TVP) representa la fracción de píxeles correctamente detectados, ecuación (4), tasa de falsos positivos (TFP), ecuación (5) es la fracción de píxeles detectados erróneamente como parte del cetáceo tomando en cuenta el fondo de la imagen. La precisión (ACC), la ecuación (8), se mide por la relación entre el número total de píxeles clasificados correctamente (suma de los verdaderos positivos y verdaderos negativos) por el número de píxeles de la imagen. Sensibilidad (SN) refleja la capacidad de un algoritmo para detectar los píxeles, ecuación (7). Especificidad (SP) es la capacidad de detectar píxeles no pertenecientes al objeto a segmentar, se puede expresar como la ecuación (6). El valor predictivo positivo (VPP) o la tasa de precisión, da la proporción de píxeles identificados que son verdaderos [6, 7], ecuación (8). Las imágenes de color de los cetáceos presentan una dimensión arbitraria ya que los investigadores de estos animales no tienen estandarizadas las medidas. A continuación se presentan los resultados obtenidos por este método propuesto de segmentación.

2.3. Experimento para el método propuesto

Las tablas 1, 2 y 3 presentan los resultados comparativos para las imágenes adquiridas a través de la base de datos proporcionada por el CICIMAR que consta de 750 fotografías. En la tabla 1, se presenta los resultados obtenidos a partir del método propuesto para la imagen segmentada #1 “Aleta ganchuda”. La Figura 6 muestra los resultados obtenidos de este “Aleta ganchuda”. En adición se presenta las imágenes e histogramas obtenidos a partir de este método.

Tabla 1. Resultados para la imagen “Aleta ganchuda” con el método propuesto.

Criterio	Método propuesto
Especificidad(%)	84.79
Sensibilidad(%)	70.0
Precisión(%)	84.79

Tabla 2. Resultados para la imagen “Aleta falcada clara” con el método propuesto.

Criterio	Método propuesto
Especificidad(%)	73.49
Sensibilidad(%)	98.90
Precisión(%)	98.90

Tabla 3. Resultados para la imagen “Aleta falcada striped” con el método propuesto.

Criterio	Método propuesto
Especificidad(%)	37.89
Sensibilidad(%)	96.90
Precisión(%)	96.90

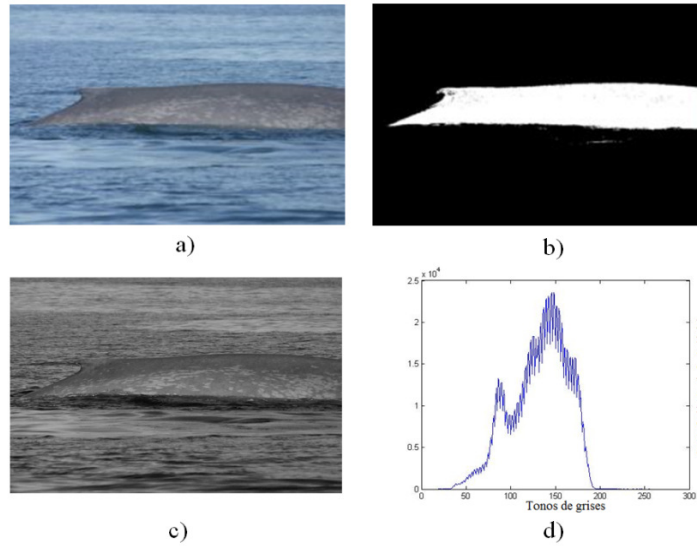


Fig. 6. Experimento 1. a) Imagen original “Aleta ganchuda”, b) imagen segmentada, c) imagen original en escala de grises, d) histograma de la imagen a segmentar.

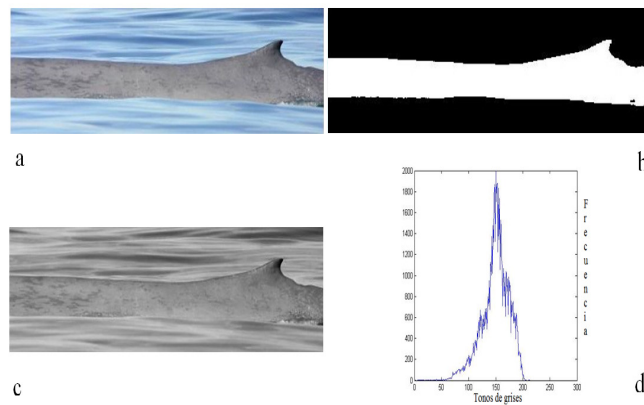


Fig. 7. Experimento 2. a) Imagen original “Aleta clara falcada”, b) imagen segmentada, c) imagen original en escala de grises, d) histograma de la imagen a segmentar.

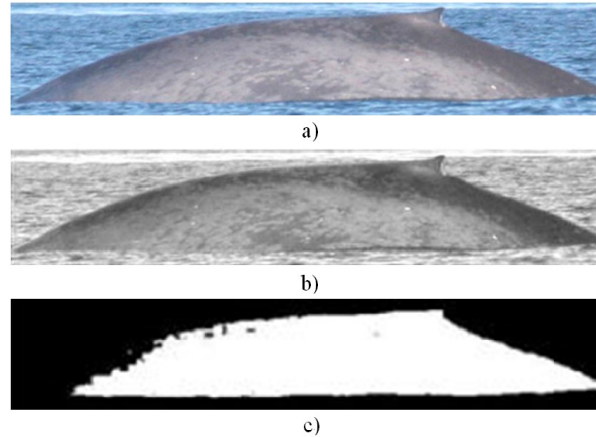


Fig. 8. Experimento 3. a) Imagen original “Aleta falcada striped”, b) imagen segmentada, c) imagen original en escala de grises, d) histograma de la imagen a segmentar.

3. Discusión

El algoritmo aquí propuesto para la segmentación de imágenes reales, dio resultados satisfactorios. Estos servirán para poder identificar el tipo de aleta de la ballena azul. Este algoritmo se irá perfeccionando para que la segmentación llegue a tener una eficiencia mucho más grande que la expuesta en este trabajo. De los resultados experimentales expuestos en este artículo, se puede ver que el método propuesto provee una mejor calidad en la segmentación de imágenes reales, explícitamente de los cetáceos en su hábitat natural. En los resultados visuales subjetivos mostrados en las Figuras 6, y 7, se puede observar que las imágenes segmentadas conservan las características principales de la forma de su aleta, siendo esto de utilidad para finalmente realizar la identificación de estas en un sistema autónomo. Siendo la Figura 8 la más baja en cuestión de medida de sensibilidad se refiere. El algoritmo tendrá que ir mejorando en el avance de la investigación para mejorar las condiciones ruidosas que son naturales en medios naturales en donde se desenvuelven los cetáceos.

4. Conclusión

El método propuesto para la segmentación de imágenes usa la desviación media aplicada en la componente de aproximaciones de la DWT, para proveer una medida de especificidad, sensibilidad y precisión mucho mayor mostrada en el trabajo [2].

El umbral basado en la desviación media mejora la técnica de segmentación [2] preservando así los detalles finos en las imágenes tratadas. Aún falta detalles a considerar para lograr finalmente una sensibilidad superior obtenida en la Tabla 3 de este trabajo.

Agradecimientos. El equipo de trabajo agradece al IPN y a CONACYT en general por el soporte de esta investigación.

Referencias

- 1 Carvajal-Gómez, B.E., Gallegos-Funes, F.J., López-Bonilla, J.L.: Método simple de ocultamiento de datos para imágenes RGB basado en Estimación de la Varianza. COMIA (2012)
- 2 Carvajal-Gómez, B.E., Gallegos-Funes, F.J., Casas-Pérez, C., Muñoz-Pérez, C., Reyes-Morales, A., Gendron, D., Hernández-Pérez, A., Algoritmo de segmentación de imágenes reales de la ballena azul en el mar de Cortes. CNIES (2012)
- 3 Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M.: Detection of Blood Vessels in Retinal Images Using Two-Dimensional Matched Filters. IEEE Transactions On Medical Imaging, vol. 8, No. 3, pp. 263-269, September (1989)
- 4 Croll, D.A., Gendron, D.: An integrated approach to the foraging ecology of marine birds and mammals. Deep-Sea Research II, vol. 45, pp: 1353-1371, 1998
- 5 De la Rosa-Flores, R.: Procesamiento de imágenes digitales. Semana de la informática, Instituto Tecnológico de Puebla, (2007)
- 6 Fathi, A., Naghsh-Nilchi, A.R.: Automatic wavelet-based retinal blood vessels segmentation and vessel diameter estimation. Biomedical Signal Processing and Control. vol. 8, pp: 71–80, Enero (2013)
- 7 Fraz, M.M., Barman, S.A., Remagnino, P., Hoppe, A., Basit, A., Uyyanonvara, B., Rudnicka, A.R., Owen, C.G.: An approach to localize the retinal blood vessels using bit planes and centerline detection. Computer Methods and Programs in Biomedicine, vol. 108, pp: 600–616, Noviembre (2012)
- 8 Gendron, D., Ugalde-de la Cruz A.: A new classification method to simplify blue whale photoidentification technique. Journal of Cetacean Research and Management, vol. 12, pp: 79-84 (2012)
- 9 Kumar, A., Kumar-Gaur, A., Srivastava, M.: A Segment based Technique for detecting Exudate from Retinal Fundus image. Procedia Technology, vol. 6, pp:1–9 (2012)
- 10 Livens, S., Scheunders, P., Van de Wouwer, G., Van Dyck D.: Wavelets for Texture Analys. Image processing and its applications, vol. 2 , pp:581-585 (1997)
- 11 Marín, D., Aquino, A., Gegúndez-Arias, M.E., Bravo, J.M.: A New Supervised Method for Blood Vessel Segmentation in Retinal Images by Using Gray-Level and Moment Invariants-Based Features. IEEE Transactions On Medical Imaging, Vol. 30, pp:146-158 (2012)
- 12 Mendonca, A.M., Campilho, A.: Segmentation of retinal blood vessels by combining the detection of centerlines and morphological.

Segmentación de peatones a partir de vistas aéreas

Jorge Ivan Rivalcoba Rivas¹ y Isaac Rudomín²

¹ Tecnológico de Monterrey, Campus Estado de México,
México

² Barcelona Supercomputing Center,
Barcelona, España

Resumen La simulación de multitudes conducida por datos ha tomado gran importancia en años recientes debido a que permite atribuir comportamientos complejos a las multitudes no logrados por otras técnicas. La información del comportamiento de personas en una multitud es extraída de secuencias de vídeo; para ello es necesario segmentar de manera eficiente a las personas en la escena, los algoritmos desarrollados en el estado del arte están orientados a la segmentación de personas empleando tomas con vistas laterales, las cuales presenta problemas de oclusión, siendo éstas de poca utilidad para el análisis de multitudes. En el presente trabajo se propone un método de segmentación de personas específicamente para vistas aéreas, para ello se emplean como descriptores los Histogramas de gradientes orientados (HOG) en conjunto con un grupo de clasificadores SVM, la novedad de la presente propuesta es la división de las tareas de clasificación condicionadas con la posición de la persona en la escena. Los resultados han permitido la segmentación de personas en vista aérea en tiempo real, la información obtenida de la etapa de segmentación sera usada para alimentar un sistema de seguimiento mismo que generara las trayectorias de los personajes en la escena que serán utilizadas por un simulador de multitudes.

Keywords: vista aérea, peatones, segmentación.

1. Introducción

La simulación de multitudes por computadora ha progresado de manera significativa desde su concepción ya hace más de dos décadas atrás. En los filmes, videojuegos y en todo tipo de mundos virtuales hemos podido apreciar la simulación de toda clase de multitudes, desde dinosaurios, aves, zombis hasta peatones escapando de desastres naturales. A pesar de que estas multitudes difieren en tamaño y comportamiento, todas en conjunto persiguen un mismo objetivo «*simular una multitud lo mas realista posible a un bajo coste computacional*».

Las multitudes han sido estudiadas desde hace tiempo atrás, como por ejemplo [1] realizo diversos estudios de multitudes durante la revolución francesa, este trabajo fue seguido por [2] el cual centro sus esfuerzos en escenas de pánico,

resultados que fueron aplicados por [3] en teoría de juegos, dichos estudios permitieron deducir que el comportamiento de una multitud no puede ser explicado por el promedio de las acciones de los individuos.

Entre las tareas que más han tenido interés en la simulación de multitudes son:

1. Apariencia.
2. Comportamiento.

La apariencia también llamada «**Rendering**» involucra a todas las técnicas que tienen como objetivo principal ejecutar las tareas de dibujado usando de manera eficiente los recursos computacionales, estas tareas se vuelven primordiales cuando el tamaño de la multitud a simular implica millones de agentes, además de lo anterior, la apariencia también persigue como meta principal que la estética de la multitud provea de la suficiente variedad de personajes permitiendo que esta sea vea más apegada a la realidad.

Por el otro lado, el comportamiento busca que los movimientos de los agentes en la escena se vean lo más plausibles e inteligentes, los trabajos más importantes que atienden la parte cognitiva de los personajes en una simulación suelen caer en cualquiera de estas categorías [4]:

- Autómatas celulares
- Fuerzas sociales
- Sistemas basados en reglas

En cuanto a los sistemas basados en reglas existen los métodos:

1. *Directos*: Basados en reglas.
2. *Indirectos*: Conducidos por datos.

Como ejemplo de simulación basada en reglas se tiene el de [5], el cual enumera un conjunto de reglas simples, que combinadas dotan de comportamientos complejos a los agentes de una multitud virtual, el problema de los sistemas basados en reglas recae en que:

- Requieren un ajuste fino.
- No simulan las variaciones sutiles que se ven en multitudes reales.
- Simulan comportamientos limitados.

Una de las soluciones que se le ha dado a estos problemas es la simulación de multitudes conducida por datos «**Data Driven**», de entre los primeros trabajos que se han presentado donde se utiliza la simulación conducida por datos está el de [6], el cual para obtener los datos que alimentarán la simulación, utiliza una cámara montada en la parte superior de la escena, donde por medio de algoritmos de visión, se realiza el registro del seguimiento de todos los peatones que cruzan un área libre, a partir de esas tomas se generan vectores de posiciones que son agrupadas mediante algoritmos de AI no supervisados, un campo de vectores extrapolado es generado por cada clase o agrupación, todos estos vectores son

usados por un simulador basado en física, cabe destacar que el seguimiento se realiza de manera semiautomática, teniendo la necesidad de la intervención de una persona para la etapa del registro de las trayectorias de cada agente en las tomas de vídeo.

En [7] se presenta otra técnica de simulación de multitudes donde los personajes virtuales exhiben comportamientos que imitan los de humanos reales, para ello se construye un espacio vectorial compuesto de estados \mathbf{S}_i y dinámica de agentes \mathbf{a}_i , donde los estados \mathbf{S}_i representan el movimiento de los agentes vecinos, el ambiente, y el movimiento propio del agente, la dinámica \mathbf{a}_i representa un vector bidimensional que corresponde a la velocidad instantánea y la dirección de movimiento del agente en cuestión.

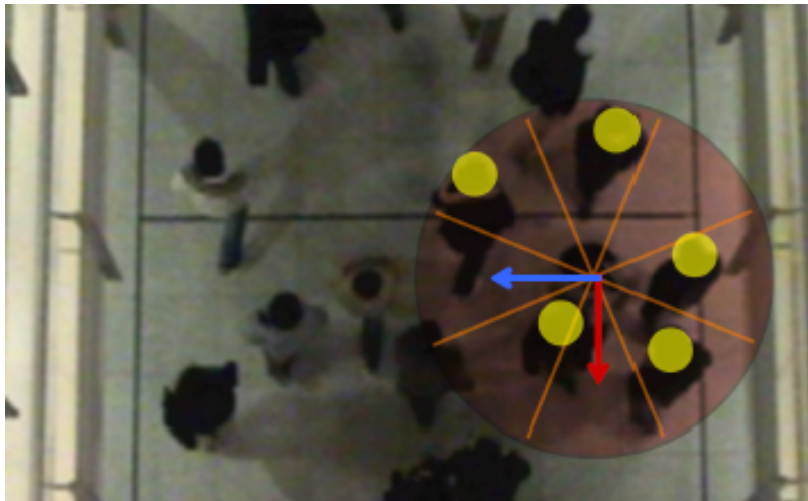


Figura 1. En [7] Para lograr capturar cada uno de los estados, se utiliza un círculo radial en el cual se miden la influencia de los vecinos sobre el propietario del círculo.

[8] utiliza algoritmos de visión y define un área de medición alrededor de cada individuo capturado en una escena, con esa información construye una base de datos misma que se usa para buscar el estado más cercano de un agente virtual, en la figura 2 se presenta el proceso mediante el cual se le atribuyen comportamientos a los agentes.

Todos los métodos de simulación previamente presentados y en general los que son conducidos por datos, obtienen la información de comportamiento de secuencias de vídeo, siendo este un proceso arduo que se incrementa proporcionalmente con el número de personas en la escena.

Para aliviar la fase de recolección de datos los investigadores han empleado algoritmos de visión que asisten la recolección de las trayectorias de cada persona en la escena. En trabajos como el de [9] se realiza la identificación y seguimiento de personas con bastante éxito, sin embargo, debido al ángulo de la toma se

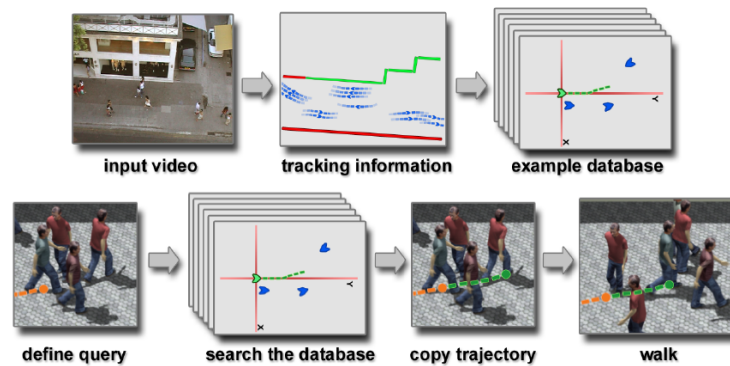


Figura 2. Cada una de las etapas presentadas por [8] mediante el cual se logra una simulación conducida por datos a través de buscar en una base de datos la escena más parecida de acuerdo a las condiciones al rededor del agente que se desea simular.

pierde la información espacial de la posición de las personas en la escena, mismo problema presenta el algoritmo presentado por [10] el cual de manera exitosa logra el conteo de personas en vista lateral pero no recaba ninguna información con respecto a la posición espacial de las personas en la escena, es por esto que las tomas aéreas resultan ser las más adecuadas en la fase de recolección de datos ya que disminuyen al máximo las oclusiones con respecto a las tomas laterales y aumenta la información en cuanto a las interacciones de las personas en las tomas, si bien los trabajos de reconocimiento y seguimiento de personas en videos es un tema bastante trabajado, generalmente no ha sido dirigidos totalmente al análisis de multitudes, existen en el estado del arte trabajos cuya finalidad es la de manejar oclusiones en vistas laterales como en [11] sin embargo el error en la detección de oclusiones suele incrementarse de manera proporcional con el número de personas en la escena, resultando estas técnicas no aptas para ser usadas escenas altamente pobladas. Se han empleado métodos semi-automáticos como el trabajo de [12] donde se emplean marcadores de colores para ayudar al sistema de visión con el seguimiento.

1.1. Objetivos

El presente trabajo trata sobre la detección de personas en vistas aéreas con el objetivo de medir las fuerzas sociales entre ellas, todo de manera automática, la información recabada servirá para que en una etapa posterior sirva para un sistema seguimiento mismo que alimentara un sistema de simulación de multitudes que dote a los agentes de comportamiento grupal e individual. Para ello el sistema se divide en dos fases:

1. Detección de los objetos de interés: En esta primer etapa se elimina el fondo de la escena para dejar tan solo los objetos de interés.

2. Detección de personas: En esta segunda etapa se identifican de manera individual a las personas en la escena, posteriormente esta información sera enviada a una etapa de seguimiento y después con la información recabada se creara una base de conocimiento.

2. Eliminación de fondo

Esta es una de las etapas mas importantes ya que se encarga de eliminar una gran parte del la información que no es necesaria. Son varios los factores que deben de ser tomados en cuenta a la hora de realizar una eliminación de fondo, [13] presenta una lista de los factores que mas afectan a la eliminación de fondo. El método mas recurrido por su facilidad de implementación es la diferencia entre cuadros contiguos en la captura de vídeo, esto es logrado extrayendo el valor absoluto de la información del cuadro actual con respecto al anterior tal como es expresado en 1. A pesar de ser el método mas preciso presenta el problema de estar condicionado al movimiento, por lo que movimientos demasiado lentos por parte de los objetos de interesen provocaran que sean clasificados como parte del fondo

$$|frame_i| - |frame_i - 1| > Th \quad (1)$$

Un método mas eficiente que tan solo la diferencias de cuadros es el «Running Average» [14] el cual modela el fondo \mathbf{B}_t calculando recursivamente el promedio del valor de cada uno de los pixeles en el escena, cada pixel es clasificado comparando la diferencia entre el cuadro actual \mathbf{I}_t y el modelo del fondo \mathbf{B}_t mediante un nivel de umbral \mathbf{T} .

$$D_t = |B_t - I_t| \quad (2)$$

$$M_t(x, y) = \begin{cases} 0, & D_t(x, y) \leq T \\ 1, & D_t(x, y) > T \end{cases} \quad (3)$$

Una ventaja de este método es la ausencia de correlación espacial entre las diferentes posiciones de los pixeles, es latamente paralelizable siendo muy directa su implementación en GPU sin hembargo una de las desventajas que presenta es la de no ofrecer un método explicito para la selección del umbral \mathbf{T} siendo que este varia de escena a escena. [15] emplea una aproximación de la mediana y la varianza para realizar la clasificación de fondo con contra objetos de interés. En [16] propone un modelo basado en gaussianas para realizar extracción de fondo en presencia de sombras, esto es realizado descomponiendo la información de color en sus correspondientes componentes de brillo y cromatismo, asumiendo que la cromaticidad es constante en una sombra pero variable en su brillo. [17] emplea un método similar pero a diferencia de [16], emplea múltiples distribuciones gaussianas para describir cada pixel de la escena, estableciendo que las distribuciones con mayor peso y menor varianza perteneces al fondo por el contrario las distribuciones con bajo peso y alta varianza pertenecen a objetos de interés. La información histórica de cada pixel, $\{X_1 \dots X_t\}$, es modelada por una

mezcla de k distribuciones gaussianas, por lo que la probabilidad de observación de cada pixel queda determinado por:

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (4)$$

Entre las ventajas de modelado por multiples gaussianas tenemos el de que es posible tener un modelo de fondo multimodal pero uno de los grandes problemas de este método es la sensibilidad a sobras y cambios de iluminación. además de no especificar como establecer los valores iniciales de esas gaussianas. Para sortear los problemas presentados anteriormente y obtener un modelo paralelizable se propone un algoritmo inspirado en el trabajo de [18] modificando la parte de la actualización del fondo la cual queda condicionada al movimiento en la escena siendo que la propuesta original actualiza todo el modelo del fondo resultando esto en un consumo computacional no necesario. El modelado del fondo se realiza por medio de la cuantización y agrupación de las observaciones en un período de tiempo, las observaciones son almacenadas en forma de códigos, al conjunto de códigos se le llama libro de códigos. Los códigos son vectores en el espacio $\{R,G,B\}$. La eliminación de fondo BGS(x) para un pixel \mathbf{x} queda definida de la siguiente manera:

1. $x = (R, G, B)I \leftarrow \sqrt{R^2 + G^2 + B^2}$
2. Para todos los códigos en el libro Ψ encontrar el código \mathbf{c}_i que mas se acerque al valor de \mathbf{x} basandose en las siguientes condiciones:
 - $colordist(\mathbf{x}, \mathbf{c}_m) \leq \varepsilon_2$
 - $brillo(I, \langle I_{max}, I_{min} \rangle) = \mathbf{true}$
3. El pixel queda determinado como:
 - Objeto: Si no se encuentran el código en el libro
 - Fondo: En caso contrario

En este caso ε_2 es el umbral de detección. La distancia de comparación queda gráficamente expresada mediante la figura 4. Para mantener el fondo se ha determinado utilizar la diferencia de cuadros de modo que tan solo los pixeles estables se actualizarán como fondo, por lo que el sistema de segmentación queda robusto ante cambios de iluminación.

3. Detección de personas

El proceso de detección de personas se realiza entrenando un clasificador SVM (se emplea la librería SVM de OpenCV), para alimentar el clasificador se utiliza un descriptor creado a partir de un histograma de gradientes orientadas como el usado por [19], dada una imagen de entrada que es definida por el tamaño de un núcleo de búsqueda, se le realizan las siguientes operaciones:

1. Cada imagen generada por el núcleo de búsqueda es convertida a escala de grises.

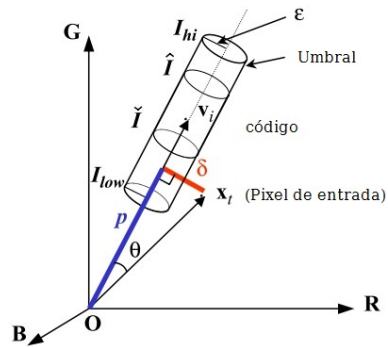


Figura 3. La evaluación de la diferencia de un código contra el pixel x_t .

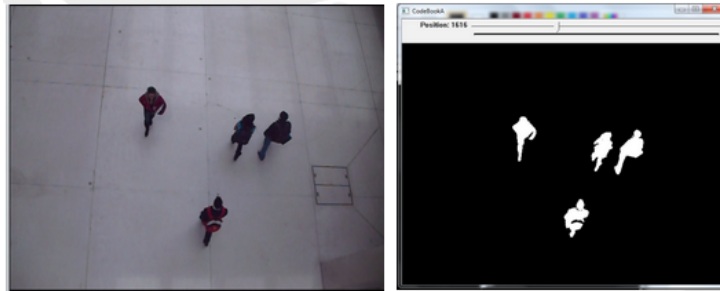


Figura 4. Resultados obtenidos con la etapa de extracción de fondo, del lado izquierdo la imagen original, de lado derecho los objetos de interés.

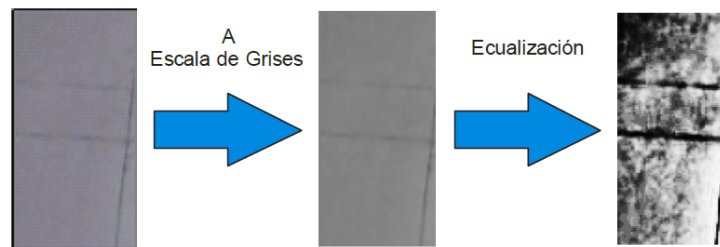


Figura 5. Procesamiento realizado al núcleo de búsqueda.

2. Se le realiza una ecualización con la finalidad de robustecer el descriptor ante variantes en la iluminación, este proceso puede observarse en la figura número 5.

Una vez realizado el pre procesamiento, se procede el cálculo de las derivadas direccionadas en «x» y «y» tal como se muestra en la **fig 6**.

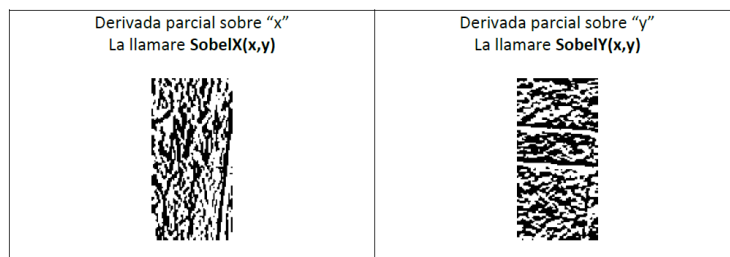


Figura 6. Procesamiento realizado al núcleo de búsqueda.

Luego se preparan en memoria arreglos de 9 imágenes vacías que servirán como los bins del histograma, cada una de estas guardara el conteo de las direcciones de los gradientes por cada pixel de la imagen original, cada una guardara un rango de ángulos, además de esto se asigna en memoria otras 9 imágenes la cuales almacenarán la integral de las imágenes anteriores: Por cada pixel de la imagen se calcula el gradiente y la magnitud de dicho gradiente mediante las siguientes ecuaciones:

$$gradiente = atan \left(\frac{SobelY(x, y)}{SobelX(x, y)} \right) * \frac{180}{\pi} + 90 \quad (5)$$

$$magnitud = \sqrt{sobelX(x, y)^2 + sobelY(x, y)^2} \quad (6)$$

Y el lugar donde sera guardada la imagen en el grupo de **bines** estará en función del valor del ángulo del gradiente, por ejemplo si se esta evaluando en $x = 0$ y $y = 0$, y el gradiente da 18° con una magnitud de 30, en este caso se tomara el la imagen cuyos bins correspondan al ángulo de entre 0° - 20° guardando en la posición $(0, 0)$ de dicha imagen la cantidad de 30. Así se recorre toda la imagen, produciendo el histograma de ángulos y magnitudes de la imagen. Con esto se genera un vector de características los suficientemente robusto para clasificar la forma humana, este algoritmo presentado en su forma original esta diseñado para detectar personas en secuencias de video, en el presente trabajo realizó una modificación al algoritmo para que permitiera detectar las cabezas para ello:

- Se ajustaron los tamaños de los núcleos de detección a 32×32 .
- Se realizó una modificación en la generación del **HOG** para obtener histogramas de $0 - 360$ grados.

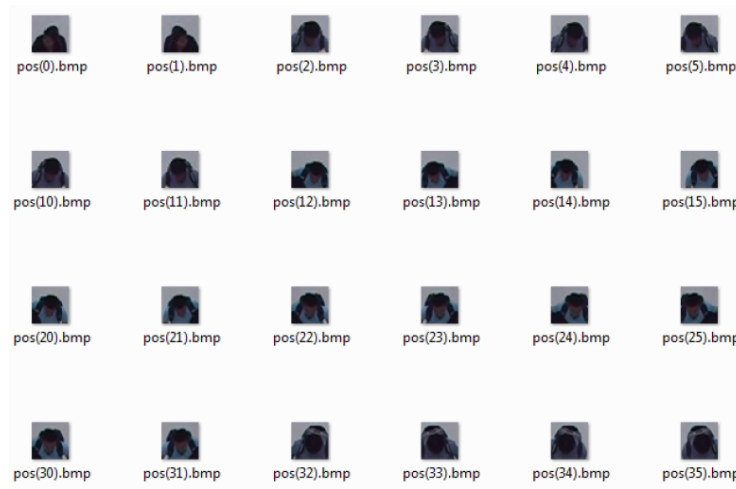


Figura 7. Imágenes que muestran el conjunto de datos para entrenar el clasificador SVM con muestras positivas

Se generó así mismo un conjunto de 1000 imágenes de entrenamiento, tanto positivas como negativas figura 7. Una vez aislado el fondo, se segmentan los objetos de interés y por cada uno de ellos se recorre un núcleo de búsqueda que usa el clasificador SVN, permitiendo clasificar cada una de las muestras en cabezas o no cabezas. Se ha determinado seleccionar a las cabezas como el elemento a seguir debido a que permanece inmóvil a lo largo del seguimiento. La granularidad de búsqueda del núcleo clasificador se muestra en la figura número 9.

4. Seguimiento

Para el seguimiento se ha utilizado el método de Lucas-Kanade que integra OpenCV. El algoritmo de seguimiento Lucas-Kanade [20] fue originalmente propuesto en el año de 1981, este método utiliza información local derivada de una ventana de acción sobre el punto de interés, siendo este el principal diferenciador con respecto a otros métodos de seguimiento, las desventajas que presenta este algoritmo de seguimiento es que debido al uso de información local, no da buenos resultados cuando los movimientos en el objeto de interés llegan a ser demasiado amplios, para mejorar este aspecto se procedió a desarrollar una implementación piramidal del método, permitiendo empezar con ventanas sobre una versión sub muestreada de la imagen hasta llegar a la imagen original. Otro aspecto importante en la detección de la cabeza fue contemplar la información de la deformación de la figura humana la cual se incrementa en la medida en que el objeto de interés se aleja de la cámara figura 10. Para ello se realizó un entrenamiento generando diferentes clasificadores los cuales entrarían de acuerdo al mapa de calibración de la escena mostrado en la figura 11.

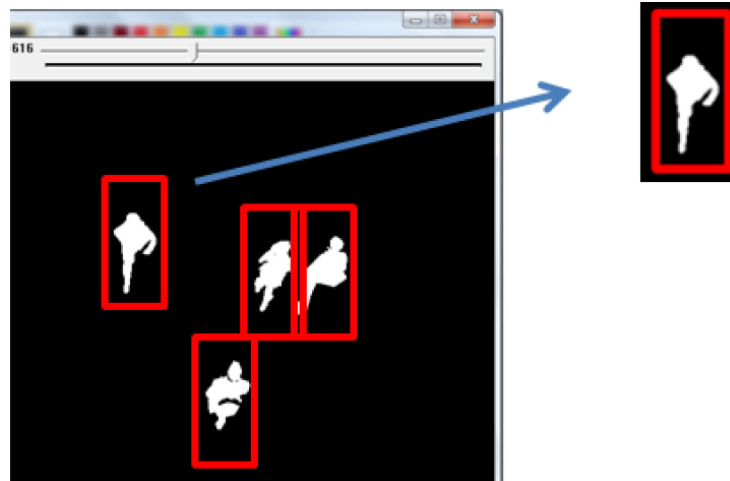


Figura 8. Se realiza una detección de blobs y sobre cada uno de ellos se recorre un núcleo de detección de cabezas para poder realizar el seguimiento

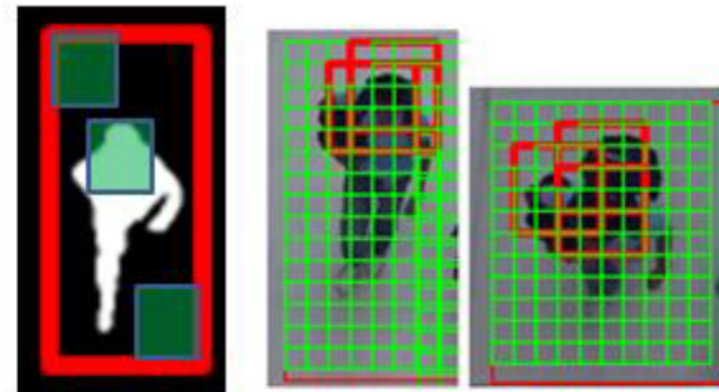


Figura 9. En esta imagen se muestra la granularidad del núcleo de detección, este recorrido se realiza por cada blob identificado en la etapa de segmentación



Figura 10. En esta imagen se muestra el grado de deformación que presenta una persona cuando se incrementa su distancia con respecto a la posición de la cámara

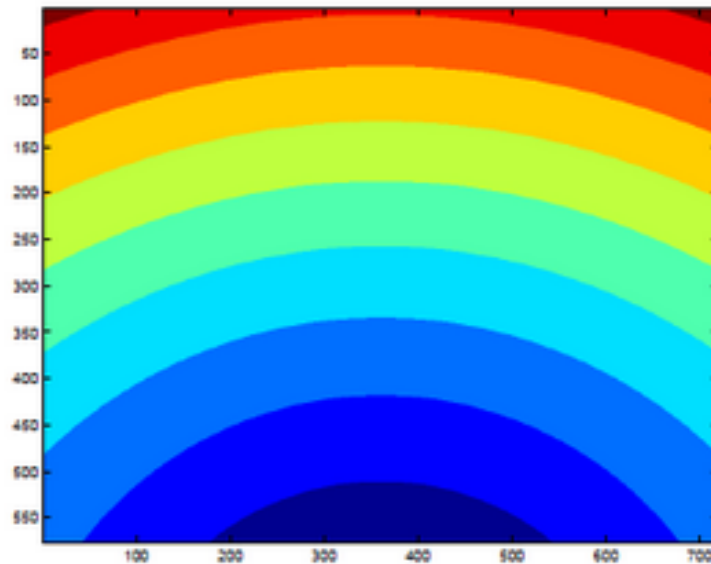


Figura 11. En esta imagen se muestra el mapa que condiciona los núcleos de detección, para cada segmento de color se aplica un núcleo de detección diferente.

5. Conclusión

Se ha presentado una metodología que permite la segmentación de personas en vista aérea, con el fin de utilizar dicha información para obtener la interacción social de un grupo de personas en una secuencia de vídeo, si bien hay una gran variedad de algoritmos que garantizan la correcta segmentación de personas, estos están diseñados principalmente para vistas laterales, las tomas laterales no son de gran utilidad para el análisis de multitudes ya que reducen en gran porcentaje la información espacial de cada uno de los individuos en la escena. Para la segmentación de personas se han utilizado dos etapas, la primera de ellas elimina el fondo de la escena y deja los objetos de interés, que en este caso son las personas, seguido de eso a cada objetos de interés se le aplica un núcleo de reconocimiento cuya finalidad es detectar las personas en la escena, se emplearon como descriptores los histogramas de gradientes orientados los cuales han probado ser efectivos a la hora de segmentar la figura humana. En trabajo futuro se pretende paralelizar cada una de las fases del sistema aquí presentado para incrementar la velocidad a la cual se obtienen los resultados.

Referencias

1. Le Bon, G.: The Crowd Study of Popular Mind by Gustave Le Bon. CreateSpace Independent Publishing Platform (2013)

2. Michalewicz, Z., Fogel, D.B.: How to Solve It: Modern Heuristics (2000)
3. Teitelbaum, P., Epstein, A.: This Week's Citation Classic. *Psychol. Rev* (7) (1962) 1981
4. Charalambous, P., Chrysanthou, Y.: Learning Crowd Steerin Behaviors from Exampales. In And, R.B., And, Y.C., Komura, T., eds.: *Motion in Games - Third International Conference*. Volume 6459., Springer (2010) 35
5. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics* **21**(4) (August 1987) 25–34
6. Musse, S.R., Jung, C.R., Jacques, J.C.S., Braun, A.: Using computer vision to simulate the motion of virtual agents. *Computer Animation and Virtual Worlds* **18**(2) (May 2007) 83–93
7. Lee, K., Choi, M., Hong, Q., Lee, J.: Group behavior from video: a data-driven approach to crowd simulation. In: *Proceedings of the 2007 ACM ...*, San Diego, California (2007) 109–118
8. Lerner, A.: Crowds by example. *Computer Graphics Forum* **26**(3) (September 2007) 655–664
9. Seemann, E., Leibe, B., Schiele, B.: Multi-aspect detection of articulated objects. *Computer Vision and Pattern ...* **2** (2006) 1582–1588
10. Subburaman, V.B., Descamps, A., Carincotte, C.: Counting People in the Crowd Using a Generic Head Detector. In: *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, IEEE (September 2012) 470–475
11. Nevatia, R.: Segmentation and tracking of multiple humans in complex situations. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. CVPR 2001. Volume 2., IEEE Comput. Soc (2001) II–194–II–201
12. Xiaoyu Deng Coll. of Comput. Sci., Zhejiang Univ., H., Liu, J.B.Z.Y.C.C.Y.: A block-based background model for video surveillance. In: *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE (March 2008) 1013–1016
13. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. *Proceedings of the Seventh IEEE International Conference on Computer Vision* (1999) 255–261 vol.1
14. Mayo, Z., Tapamo, J.: Background subtraction survey for highway surveillance. *Proc. of the Annual Symposium of the Pattern ...* (2009)
15. Manzanera, A., Richefeu, J.C.: A new motion detection algorithm based on Σ - Δ background estimation. *Pattern Recognition Letters* **28**(3) (February 2007) 320–328
16. Porikli, F., Tuzel, O.: Bayesian background modeling for foreground detection. *Proceedings of the third ACM international workshop ...* (2005) 55
17. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Cat. No PR00149) 246–252
18. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. *Real-Time Imaging* **11**(3) (June 2005) 172–185
19. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* **1** (2005) 886–893
20. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. *Proceedings of the 7th international joint ...* **130** (1981) 121–130

Redes neuronales y sistemas híbridos inteligentes

Clasificación de arritmias cardiacas utilizando KNN y Naive Bayes mejorada con algoritmos genéticos (AG) y optimización de cúmulo de partículas (PSO)

Christian Padilla-Navarro¹, Rosario Baltazar-Flores¹, David Cuesta-Frau² y Victor Zamudio-Rodríguez¹

¹ División de Estudios de Posgrado e Investigación, Instituto Tecnológico de León, Av. Tecnológico S/N Col. Industrial Julián de Obregón, 37290, León, México
{jocripa, r.baltazar, vic.zamudio}@ieee.org
<http://posgrado.itleon.edu.mx/>

²Instituto Tecnológico de Informática, Universidad Politécnica de Valencia, Campus Alcoi, Plaza Ferrándiz y Carbonell, 2, 03801 Alcoi, España
{dcuesta}@disca.upv.es
<http://www.epsa.upv.es/>

Resumen En la presente investigación se busca aumentar el porcentaje de clasificación de las características de una base de datos de arritmias cardiacas aplicando metaheurísticas (Algoritmos Genéticos con y sin Elitismo y Optimización por Cúmulo de Partículas) y diversos clasificadores (KNN con un vecino -1NN-, KNN con tres vecinos -3NN-, KNN con cinco vecinos -5NN- y Naive Bayes), con el fin de realizar una selección de las características principales y reducir el ruido en la base de datos. El aumento en el porcentaje de clasificación en todos los casos fue siempre mayor al 10%, y la reducción de características llegó a ser en algunos casos hasta del 79%.

Palabras clave: ECG, arritmia, KNN, Naive Bayes, AG, PSO, reducción de características.

1. Introducción

La función principal de los clasificadores es separar a través de clases los distintos datos con los que contamos. Existen diversas aplicaciones de clasificadores en señales de ECG, que tienen la finalidad de distinguir entre las clases de arritmias existentes. En Nasiri [16] se realizó la clasificación de arritmias provenientes de señales de ECG aplicando Máquinas de Soporte Vectorial (SVM) y Algoritmos Genéticos. Friedman [20] muestra distintas técnicas del Reconocimiento de Patrones, diversos clasificadores y sus variantes. En Badeeh [1] se aplicó el aprendizaje de las máquinas en el diagnóstico de ECG. Ramírez [18] propuso un modelo de clasificación dinámico de arritmias cardiacas mediante aprendizaje de máquina con interfaz a usuario. En Mohamed [22] se presentaron

dos métodos para la clasificación de arritmias multiclase aplicando el Análisis de Componentes Principales (PCA), las Máquinas de Soporte de Vectores Difusos, y el Agrupamiento Desbalanceado. Kallas [19] mostró la clasificación de arritmias multiclase a través de Máquinas de Soporte Vectorial (SVM) combinadas con la extracción de características a través del Análisis de Componentes Principales en señales de ECG. En Thanapatay [8] se propuso un nuevo método de clasificación para ECG aplicando el Análisis de Componentes Principales (PCA) y Máquinas de Soporte Vectorial (SVM). Rabee [4] presentó la clasificación de señales de ECG utilizando Máquinas de Soporte Vectorial y basándose en el análisis de Wavelets de Multiresolución. En Shen [21] se propuso un modelo de clasificación aplicando Máquinas de Soporte Vectorial (SVM) y el Análisis de Componentes Independientes y Zellmer [9] logró la clasificación de señales de ECG basándose en la transformada Wavelet continua y Máquinas de Soporte Vectorial (SVM).

La combinación de clasificadores con metaheurísticas para la reducción de características en las señales de ECG es una forma eficiente de eliminar datos innecesarios o que generan ruido, ayudando así en la búsqueda de anomalías en los latidos. En Martínez [3] se propone un modelo de clasificación de ECG aplicando sistemas inteligentes para la detección de problemas del corazón aplicando Redes Neuronales y Algoritmos Genéticos. En Melgani [11] se realizó la clasificación de señales de ECG a través de Máquinas de Soporte Vectorial (SVM) aplicando el algoritmo PSO (Particle Swarm Optimization). Mientras que en Daamouche [2] se buscó optimizar la clasificación aplicando el algoritmo de PSO, las Máquinas de Soporte Vectorial (SVM) y realizando la reducción de ruido en las señales a través de Wavelets. En Fira [23] se investigaron los resultados de la clasificación de la compresión de señales de ECG basándose en diversos tipos de matrices de proyección utilizando el algoritmo KNN (K-Nearest Neighbour). En Vaish [7] se investigan las cargas de eficacia y eficiencia computacional de diferentes algoritmos que se utilizan para reconocer el estado emocional a través de señales fisiológicas cardiovasculares, utilizando tablas de decisión, Perceptrón Multicapa, C4.5 y Naive Bayes como un objeto de estudio, la clasificación la realizaron en dos ámbitos: la excitación alta y baja.

La capacidad de identificar automáticamente las arritmias de ECG es importante para el diagnóstico clínico y el tratamiento. En Soman [24] se utilizaron sistemas de aprendizaje automático, Oner, J48 y Naive Bayes para clasificar conjuntos de datos de arritmias obtenidas de ECG médicos. En Gao [6] se describe un sistema para la detección de arritmias cardíacas en señales ECG, basándose en una red neuronal artificial bayesiana (ANN). El clasificador ANN se construye mediante el uso de un modelo de regresión logístico y el algoritmo BackPropagation.

2. Datos de prueba

Para realizar las pruebas utilizamos el conjunto de datos de arritmias del UCI [10] (Center for Machine Learning and Intelligent System). Esta Base de Datos contiene 279 atributos, de los cuales 206 fueron evaluados linealmente y el resto de forma nominal. Dichos datos están basados principalmente en las ondas P, Q, R, S, T y U, y los segmentos que se forman con estas en diversos canales. La base de datos contiene 452 instancias y algunos valores perdidos.

2.1. Función objetivo

Inicialmente se realizó la clasificación de la base de datos con cada uno de los clasificadores propuestos (1NN, 3NN, 5NN y Naive Bayes), obteniendo como función objetivo el porcentaje de clasificación. Posteriormente, se propuso aplicar diversas Metaheurísticas (Algoritmos Genéticos, Genéticos con Elitismo y PSO), siempre sin perder de vista el porcentaje de clasificación como el objetivo principal. En esta investigación no fue considerada la reducción de características como objetivo, pero si como auxiliar en la búsqueda de la mejor clasificación. En todos los casos la selección de características se realizó de manera aleatoria.

3. Algoritmos genéticos (AG)

Formalmente, y de acuerdo a la definición de Goldberg, "Los Algoritmos Genéticos son algoritmos de búsqueda basados en la mecánica de la selección natural y de la genética natural. Combinan la supervivencia con el mejor individuo entre las estructuras de secuencia, con la posibilidad del intercambio de información estructurada, muchas veces de forma aleatoria, para formar un algoritmo de búsqueda que tiene algunas veces la forma de búsqueda que utilizada por los humanos" [14].

3.1. Operadores genéticos utilizados

Selección. El proceso de selección se realizó a través del método Vasconcelos. Para aplicar este método necesitamos ordenar la aptitud de todos los individuos, ascendente o descendentemente, y tomar el individuo más apto y el menos apto.

Cruza. Realizamos la cruce a partir de dos puntos aleatorios. Se toma de la cadena del Padre (mejor individuo) desde la posición 0 hasta el primer punto aleatorio, desde el primer punto aleatorio hasta el segundo punto aleatorio se toma de la cadena de datos de la Madre (peor individuo), y finalmente desde el segundo punto hasta terminar la cadena se obtiene del Padre.

Muta. Se muta un porcentaje de la población, se toma un dato de la cadena de manera aleatoria y se cambia su valor de 0 a 1 o de 1 a 0.

Elitismo. Se toma el mejor individuo y se clona un porcentaje de veces.

En el **Algoritmo 1** se muestra el Algoritmo Genético Básico utilizado en el proceso de clasificación. Se ha demostrado que el elitismo es fundamental en los algoritmos genéticos, en el caso del Algoritmo Genético con Elitismo se realiza la clonación del mejor individuo en un porcentaje de la población.

Algoritmo 1 Algoritmo Genético Simple para la Clasificación de ECG

- 1: Datos de entrada: tamaño de la población, porcentaje de mutación, llamadas a función, función objetivo -clasificador-(1NN, 3NN, 5NN, Naive Bayes), número de capas.
 - 2: Inicializar una población aleatoria.
 - 3: Evaluar la aptitud de cada uno de los individuos (porcentaje de clasificación).
 - 4: **mientras** (no concluyan las llamadas a función) **hacer**
 - 5: Seleccionar al mejor y al peor individuo (Vasconcelos).
 - 6: Realizar la cruce entre los padres en un bit aleatorio.
 - 7: Mutar un porcentaje de descendientes.
 - 8: Evaluar la aptitud de cada individuo (porcentaje de la clasificación).
 - 9: Mostrar el mejor individuo.
 - 10: **fin mientras**
-

4. Optimización por cúmulo de partículas (PSO)

PSO es una metaheurística inspirada en la conducta social de de partículas, comunmente aplicada para la solución de problemas de optimización. En 1995 Kennedy y Eberhart [17] desarrollaron el primer algoritmo. El algoritmo puede utilizarse en funciones continuas o binarias.

La actualización de las velocidades de las partículas puede ser vista en la **Ecuación (1)**.

$$v_i = wv_i + \phi_1(GBest_i - x_i) + \phi_2(LBest_i - x_i) \quad (1)$$

Mientras tanto, para realizar la actualización del valor de x_i se emplea una sigmoide, representada con la **Ecuación (2)**.

$$\overrightarrow{Sig}(V_{ij}) = \frac{1}{1 + \exp^{-v_{ij}}} \quad (2)$$

Algoritmo 2 Algoritmo PSO para la Clasificación de ECG

```

1: Datos de entrada:  $\phi_1[0, 1]$ ,  $\phi_2[0, 1]$ , tamaño de la población, llamadas a función, función
   objetivo -clasificador-(1NN, 3NN, 5NN, Naive Bayes), número de capas.
2: Inicializar una población aleatoria.
3: Evaluar la aptitud de cada individuo (porcentaje de clasificación).
4: Tomar al mejor individuo y la mejor aptitud (GBest).
5: Guardar la primera aptitud de cada individuo (LBest).
6: Generar velocidades aleatorias  $V[0, 1]$ .
7: mientras (no concluyan las llamadas a función) hacer
8:   para cada partícula i hacer
9:     para cada miembro de la partícula j hacer
10:      Actualizar velocidades (ver Ecuación 1).
11:      Actualizar el valor de  $x_i$  (ver Ecuación 2).
12:     fin para
13:     para cada miembro de la partícula j hacer
14:      Generar un número aleatorio  $r_{ij}[0, 1]$ .
15:      si  $r_{ij} < Sig(V_{ij})$  entonces
16:         $x_{ij} = 0$ 
17:      fin si
18:      si  $r_{ij} > Sig(V_{ij})$  entonces
19:         $x_{ij} = 1$ 
20:      fin si
21:     fin para
22:     Evaluar la aptitud de cada individuo (porcentaje de clasificación).
23:     Actualizar el LBest.
24:     si  $f(x_i) > f(LBest_i)$  entonces
25:        $LBest_i = x_i$ 
26:     fin si
27:     fin para
28:     Encontrar el mejor LBest
29:     si  $LBest > GBest$  entonces
30:        $GBest_i = LBest_i$ 
31:     fin si
32: fin mientras

```

5. Experimentos y resultados

Se realizaron diversas pruebas, todas para 2000 llamadas a función y 10 capas. En el caso de los Algoritmos Genéticos, el porcentaje de mutación utilizado fue del 20% y 10% de elitismo, dichos porcentajes fueron propuestos a manera experimental. En el caso de PSO, $\phi_1 = 0.3$, $\phi_2 = 0.5$ y $w = 0.7$. Las primeras pruebas se hicieron con los clasificadores, 1NN, 3NN, 5NN y Naive Bayes, sin aplicar metaheurísticas. Se consideraron todas las características para tener un parámetro de arranque y poder comparar buscando una mejora. Posteriormente,

se probó con los mismos clasificadores, pero se aplicó el Algoritmo Genético, el Genético con Elitismo y el PSO.

Se puede ver la Media del Porcentaje de Clasificación obtenido (ver la Tabla 1) y la Media del Número de Características empleadas en la clasificación (ver la Tabla 2).

Tabla 1. Media del porcentaje de clasificación.

Algoritmo	Clasificador	Población	Media
Sin algoritmo	1NN	-	52.87 %
Algoritmo Genético	1NN	10	66.27 %
Genético con Elitismo	1NN	10	65.35 %
PSO	1NN	10	64.49 %
Algoritmo Genético	1NN	100	63.41 %
Genético con Elitismo	1NN	100	62.92 %
PSO	1NN	100	64.78 %
Sin algoritmo	3NN	-	57.74 %
Algoritmo Genético	3NN	10	67.71 %
Genético con Elitismo	3NN	10	66.77 %
PSO	3NN	10	67.35 %
Algoritmo Genético	3NN	100	65.71 %
Genético con Elitismo	3NN	100	65.21 %
PSO	3NN	100	67.42 %
Sin Algoritmo	5NN	-	59.29 %
Algoritmo Genético	5NN	10	67.40 %
Genético con Elitismo	5NN	10	64.20 %
PSO	5NN	10	67.28 %
Algoritmo Genético	5NN	100	65.14 %
Genético con Elitismo	5NN	100	65.21 %
PSO	5NN	100	67.35 %
Sin Algoritmo	Naive Bayes	-	61.72 %
Algoritmo Genético	Naive Bayes	10	67.61 %
Genético con Elitismo	Naive Bayes	10	66.41 %
PSO	Naive Bayes	10	68.42 %
Algoritmo Genético	Naive Bayes	100	65.55 %
Genético con Elitismo	Naive Bayes	100	65.27 %
PSO	Naive Bayes	100	67.57 %

Tabla 2. Media del número de características empleadas para la clasificación.

Algoritmo	Clasificador	Población	Media
Sin Algoritmo	1NN	-	279
Algoritmo Genético	1NN	10	138.87
Genético con Elitismo	1NN	10	141.83
PSO	1NN	10	85.26
Algoritmo Genético	1NN	100	139.65
Genético con Elitismo	1NN	100	137.78
PSO	1NN	100	74.14
Sin Algoritmo	3NN	-	279
Algoritmo Genético	3NN	10	126.86
Genético con Elitismo	3NN	10	142.94
PSO	3NN	10	80.04
Algoritmo Genético	3NN	100	147.07
Genético con Elitismo	3NN	100	141.29
PSO	3NN	100	78.07
Sin Algoritmo	5NN	-	279
Algoritmo Genético	5NN	10	134.94
Genético con Elitismo	5NN	10	146.56
PSO	5NN	10	84.54
Algoritmo Genético	5NN	100	141.08
Genético con Elitismo	5NN	100	135.90
PSO	5NN	100	71.07
Sin Algoritmo	Naive Bayes	-	279
Algoritmo Genético	Naive Bayes	10	135.34
Genético con Elitismo	Naive Bayes	10	140.93
PSO	Naive Bayes	10	80.82
Algoritmo Genético	Naive Bayes	100	138.48
Genético con Elitismo	Naive Bayes	100	145.39
PSO	Naive Bayes	100	81.85

5.1. Prueba no paramétrica de los signos de Wilcoxon

Se utilizó la prueba no paramétrica de los signos de Wilcoxon [13] para realizar la comparativa entre los distintos algoritmos, el nivel de significancia empleado fue del 0.1.

Sin algoritmo contra algoritmo genético. Aplicando la prueba de Wilcoxon [13] para realizar la comparativa entre el método Sin Algoritmo (T^+) y Algoritmo Genético (T^-) con diferentes clasificadores, y tomando la media como parámetro de referencia (ver la Tabla 1), obtenemos que:

Dado que $T = \min(T^-, T^+) = (36, 0) = 0$ y $T_0 = 5$, podemos concluir que $T \leq T_0$, y podemos aceptar la hipótesis alternativa H_A . Como el objetivo es maximizar el porcentaje de clasificación, los resultados que se encuentran más a la derecha son los del Algoritmo Genético (T^-) y por tanto se obtienen mejores resultados que con el método Sin Algoritmo.

Algoritmo genético contra algoritmo PSO. Aplicando la prueba de Wilcoxon [13] para realizar la comparativa entre el Algoritmo Genético (T^+) y el algoritmo PSO (T^-) con diferentes clasificadores, y tomando la media como parámetro de referencia (ver la Tabla 1), obtenemos que:

Dado que $T = \min(T^-, T^+) = (10, 11) = 10$ y $T_0 = 5$, podemos concluir que no se cumple $T \leq T_0$, y no podemos aceptar la hipótesis alternativa H_A . No es posible determinar cual es el algoritmo que se encuentra más a la derecha.

5.2. Algoritmo genético contra algoritmo genético con elitismo

Aplicando la prueba de Wilcoxon [13] para realizar la comparativa entre el Algoritmo Genético (T^+) y el algoritmo Genético con Elitismo (T^-) con diferentes clasificadores, y tomando la media como parámetro de referencia (ver la Tabla 1), obtenemos que:

Dado que $T = \min(T^-, T^+) = (28, 8) = 8$ y $T_0 = 5$, podemos concluir que no se cumple $T \leq T_0$, y no podemos aceptar la hipótesis alternativa H_A . No es posible determinar cual es el algoritmo que se encuentra más a la derecha.

Algoritmo genético con elitismo contra algoritmo PSO. Aplicando la prueba de Wilcoxon [13] para realizar la comparativa entre el Algoritmo Genético con Elitismo (T^+) y el algoritmo PSO (T^-) con diferentes clasificadores, y tomando la media como parámetro de referencia (ver la Tabla 1), obtenemos que:

Dado que $T = \min(T^-, T^+) = (28.5, 7.5) = 7.5$ y $T_0 = 5$, podemos concluir que no se cumple $T \leq T_0$, y no podemos aceptar la hipótesis alternativa H_A . No es posible determinar cual es el algoritmo que se encuentra más a la derecha.

6. Conclusiones

En todos los casos, el aumento en la clasificación fue mayor al 10% utilizando una metaheurística como auxiliar en la mejora, y la reducción de características llegó hasta el 79%, lo que conllevaría un ahorro en el tiempo computacional ya que se trabajaría con una cantidad inferior de datos y, por tanto, un avance considerable en caso de ser utilizado en la clasificación en tiempo real en donde es de suma importancia obtener una solución de forma rápida. Además, podemos decir que la reducción de características nos puede llevar a un aumento en el porcentaje de clasificación y por tanto a reducir la cantidad de datos implementados.

Después de haber realizado las pruebas y aplicado la prueba no paramétrica de los signos de Wilcoxon, podemos concluir que es posible implementar una metaheurística como auxiliar en la mejora de la clasificación de señales de ECG, pero no fue posible determinar si PSO, AG o AG con Elitismo nos lleva a los mejores resultados.

Agradecimientos. Agradecemos al proyecto 4573.12-P de la DGEST por su apoyo en esta investigación, y el autor Christian Padilla-Navarro agradece al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca otorgada.

Referencias

1. Abdeel-Badeeh M. Salem, Kenneth Revett. Machine learning in electrocardiogram diagnosis. International Multiconference on Computer Science and Information Technology, 978-83-60810-14-9:429-433, 2009.
2. Abdelhamid Daamouche, Latifa Hamami, A wavelet optimization approach for ecg signal classification, Biomedical Signal Processing and Control, 7:342-349, 2012.
3. A.I.Martínez, Rojas, utilización de sistemas inteligentes para la detección de problemas del corazón mediante ecg, 2005.
4. Ayman Rabee, Ecg signal classification using support vector machine based on wavelet multiresolution analysis, The 11th International Conference on Information Sciences, Signal Processing and their Applications: Special Sessions, 978-1-4673-0382-8:1319-1323, 2012.
5. Darwin, C. On the Origin of Species by Means of Natural Selection., John Murray, 1859.
6. Dayong Gao, Michael Madden, Bayesianann classifier for ecg arrhythmia diagnostic system: A comparison study. IJCNN, 2005.
7. Dr. Abhishek Vaish, Performance analysis of machine learning algorithms for emotion state recognition through physiological signal. Global Journal of Computer Science and Technology Neural and Artificial Intelligence, 12, 2012.
8. Dusit Thanapatay, Chaiwat S., Ecg beat classification method for ecg printout with principle components analysis and support vector machines, International Conference on Electronics and Information Engineering (ICEIC), 1:72-75, 2010.
9. Erik Zellmer, Fei Shang. Highly accurate ecg beat classification based on continuous wavelet transformation and multiple support vector machine classifiers, IEEE, 978-1-4244-4134-1, 2009.
10. Frank, A. and Asuncion, A. (2010). UCI Machine Learning Repository <http://archive.ics.uci.edu/ml/datasets/Arrhythmia>. Irvine, CA: University of California, School of Information and Computer Science.
11. Farid Melgani, Classification of electrocardiogram signals with support vector machines and particle swarm optimization, IEEE Transactions On Information Technology In Biomedicine, 12:5, 2008.
12. Fogel L. J. Owens, Artificial Intelligence through Simulated Evolution, 1966.
13. Frank Wilcoxon, Individual comparisons by ranking methods.
14. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning Addison-Wesley Longman Publishing Co., 1989.
15. Holland, Adaptation in Natural and Artificial Systems, 1975.
16. Jalal A. Nasiri, Mahmoud Naghibzadeh, Ecg arrhythmia classification with support vector machines and genetic algorithm, Third UKSim European Symposium on Computer Modeling and Simulation, 978-0-7695-3886-0:187-192, 2009.
17. Kennedy J., Particle Swarm Optimization Proceedings of the 1995, IEEE international conference on neural networks, (ICNN'95), 1995, 4, 1942-1948.
18. López, Modelo de clasificación dinámico de arritmias cardiacas mediante aprendizaje de máquina con interfaz de usuario. Ciencia y Tecnología Neogranadina, 16/002:56-95, 2006.

19. Maya Kallas, Clovis Francis, Multi-class svm classification combined with kernel pca feature extraction of eeg signals. International Conference on Telecommunications, 978-1-4673-0747-5, 2012.
20. Menahem Friedman, Introduction to Pattern Recognition statical, structural, neural and fuzzy logic approaches, Imperial College Press, 1999.
21. Mi Shen, Liping Wang. Multi-lead eeg classification based on independent component analysis and support vector machine, 3rd International Conference on Biomedical Engineering and Informatics (BMEI), 978-1-4244-6498-2:960–964, 2010.
22. Mohamed Cherif Nait-Hamoud, Two novel methods for multiclass eeg arrhythmias classification based on pca, fuzzy support vector machine and unbalanced clustering, IEEE, 978-1-4244-8611-3:140-145, 2010.
23. Monica Fira, Liviu Goras, On the projection matrices influence in the classification of compressed sensed eeg signals, International Journal of Advanced Computer Science and Applications, 3, 2012.
24. Thara Soman, Classification of arrhythmia using machine learning techniques, ICOSSE, 2005.

Comparativa entre los algoritmos DE y PSO aplicada a la demodulación de interferogramas de franjas cerradas modeladas por polinomios de Zernike

Juan de Anda y Martín Carpio, Héctor J. Puga, Luis Mancilla Espinoza, Manuel Ornelas, Rosario Baltazar y Marco Sotelo

División de Estudios de Posgrado e Investigación,
Instituto Tecnológico de León,
León, Guanajuato, 37290, México
juandanda2@gmail.com, jmcarpio61@hotmail.com

Resumen Se presentan los resultados de comparación de los algoritmos de Evolución Diferencial (DE) y Particle Swarm Optimization (PSO), aplicados al problema de demodulación de interferogramas de franja cerrada modeladas con polinomios de Zernike. Se muestra la recuperación de la fase encerrada de un interferograma simulado, a partir de una función de aptitud representada como combinación lineal de polinomios de Zernike. Posteriormente se aplica la prueba de suma de rangos con signo de Wilcoxon, con el objetivo de discernir qué algoritmo presenta mejor desempeño en la demodulación de interferogramas.

Palabras clave: demodular, interferograma, polinomio de Zernike, optimización, evolución diferencial, particle swarm optimization.

1. Introducción

La Interferometría, es un conjunto de técnicas que hacen uso de la superposición de ondas electromagnéticas coherentes, con el objetivo de extraer información que se encuentra en la onda. Es una importante técnica utilizada en metrología óptica, astronomía, espectroscopía y oceanografía. Éstas buscan medir pequeños desplazamientos, cambios de índice refractivo e irregularidades en las superficies de estudio [1,2].

En el caso de metrología óptica, el modelo matemático que gobierna el patrón de interferencia tiene un comportamiento cosenoidal y está dada en [3]:

$$I(x, y) = a(x, y) + b(x, y) \cos[\Theta(x, y)]. \quad (1)$$

Donde $a(x, y)$ es la luminosidad de fondo, $b(x, y)$ es la amplitud de modulación y $\Theta(x, y)$ es la fase de la onda electromagnética.

La idea principal es recuperar la fase $\Theta(x, y)$ del patrón de franjas, que está directamente relacionada con las variables físicas del problema de estudio. La

forma en que se lleva a cabo dicha recuperación es a partir del experimento de interferencia, y que posteriormente es capturado en una imagen digital para realizar el procesamiento computacional.

Podemos separar los patrones de interferencia (Interferogramas) en dos clases, la primera contiene franjas abiertas y la segunda está constituida por franjas cerradas [1]. Para el primer caso existen técnicas basadas en el uso de la transformada de Fourier, por ejemplo, [4,5]. Sin embargo, éstos métodos sólo funcionan si tienen frecuencia portadora y la señal cuenta con poco ruido. Más aun, si dichos métodos son aplicados a interferogramas de franja cerrada tienden a fallar en el proceso de demodulación.

Al problema de demodulación de interferogramas de franja cerrada, recientemente se le ha aplicado técnicas de Soft Computing, tales como Artificial Neural Network (ANN), Genetic Algorithms (GA) y Particle Swarm Optimization (PSO). En el área de las ANN se han utilizado redes multi-capas, que son entrenadas basándose en el método del descenso de gradiente y usando los patrones de franjas [6].

Por parte de los GA se han utilizado diferentes bases polinomiales para modelar la fase. Cuevas et al. [7], utiliza una representación en base monomial de la fase para realizar la demodulación del interferograma, mientras que Mancilla et al. [8], utiliza una representación de la fase en la base ortogonal de Zernike, logrando así disminuir el tiempo en la demodulación.

Jiménez et al. [9], busca evitar la representación binaria utilizada en el GA y aplica el algoritmo PSO, el cual tiene naturaleza de ser utilizado en el dominio de los números reales. También realiza un conteo de franjas que permite disminuir el rango de búsqueda de la población inicial. Sin embargo, se presenta el problema de que las velocidades que caracterizan al PSO, escapan del rango definido para el interferograma y tienen que hacer ciertas modificaciones al algoritmo.

En el presente artículo se trata de resolver el problema de la demodulación de interferogramas, modelando la fase del interferograma con polinomios ortogonales de Zernike, utilizando los algoritmos de Evolución diferencial [10] y Particle Swarm Optimization [11]. Con el objetivo de discernir el Desempeño de los algoritmos bajo el criterio de mismo número de llamadas a función (NLF) y utilizando la prueba de suma de rangos con signo de Wilcoxon [12].

La Evolución Diferencial es un nuevo enfoque heurístico que permite minimizar o maximizar problemas de funciones de varias variables en el espacio continuo, utilizando cuatro parámetros, que son el *Peso Diferencial* (F), *Probabilidad de Cruza* (PC), *Número de Población* (NP) y el *Número de Llamadas a Función* (NLF).

El algoritmo PSO [11], fue inspirado en el comportamiento social de las parvas de pájaros y cardúmenes de peces, en cuanto a la optimización de movimientos a través del cielo y el mar respectivamente. Básicamente el algoritmo trabaja con una búsqueda de optimización local (*Local Best*) y una optimización global (*Global best*), con respecto a la función a optimizar, que se van actualizando con respecto a una nueva población en la siguiente iteración, utilizando cambios de posición dadas por restas vectoriales entre las posiciones de los mejo-

res locales y el mejor global, que es modulada por dos pesos, el *peso de suavidad local* (φ_2) y el *peso de suavidad global* (φ_1) definido en todos los reales [11]. Sin embargo, el algoritmo clásico de PSO cuenta con algunos inconvenientes a la hora de hacer los cambios de posición dados por la Ecuación (2)

$$V_{k+1} = \omega V_k + \varphi_1(G_{best} - X_k) + \varphi_2(L_{best} - X_k). \quad (2)$$

Donde ω es el peso inercial, G_{best} es el mejor global, L_{best} son los mejores locales y X_k es la posición de la partícula en la iteración k . La Ec. 2 tiende a salirse del rango inicial que fue dado para la optimización del problema, provocando así una pérdida en la convergencia del algoritmo y en el peor de los casos no encontrar la solución. Diferentes autores basándose en el problema anteriormente mencionado han presentado diferentes propuestas para solucionarlo [13,14]. Sin embargo, en el presente trabajo hemos hecho uso de la corrección a la Ecuación (2) que es presentada por Bratton and Kennedy [15], y es definida como sigue:

$$V_{k+1} = \chi(\omega V_k + \varphi_1(G_{best} - X_k) + \varphi_2(L_{best} - X_k)). \quad (3)$$

Donde χ es definida como

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}. \quad (4)$$

Y donde $\varphi = \varphi_1 + \varphi_2$, permitiéndonos la Ecuación (4) mantener regulado el cambio de posición dentro de los rangos iniciales de convergencia para nuestro problema de estudio.

La prueba de suma de rangos con signo de Wilcoxon, es una prueba no paramétrica que a partir de la comparación de la distribución de medianas de dos muestras, permite determinar si existe diferencia entre ellas [12].

2. Modelos matemáticos

El criterio general de optimización, es modelado a partir del error cuadrático que existe entre la imagen original normalizada y la imagen recuperada del interferograma, y un factor de suavizado de la primera derivada de la fase. De manera que se define como:

$$f_{opt}(z^k) = \alpha - \sum_{y=1}^R \sum_{x=1}^C \{ [I_N(x, y) - \cos[f(z^k, x, y)]]^2 + \lambda \{ [f(z^k, x, y) - f(z^k, x - 1, y)]^2 + [f(z^k, x, y) - f(z^k, x, y - 1)]^2 \} \}. \quad (5)$$

Donde x, y son valores enteros y representan el índice de localización de pixel en la imagen, $I_N(x, y)$ es la imagen normalizada entre [-1,1], α es usado para convertir el problema de minimización a maximización, y λ es el parámetro que controla el suavizado de las derivadas de la fase. Cabe mencionar que en el presente trabajo no se utilizó el término de suavizado.

La fase para el presente artículo fue modelada a partir de los polinomios ortogonales de Zernike en representación cartesiana definida como [16]:

$$f(x, y) = \sum_{j=1}^L z_j^k U_j(x, y). \quad (6)$$

Donde j es el índice de combinación lineal del polinomio de Zernike y L está dado por:

$$L = \frac{(n+1)(n+2)}{2}. \quad (7)$$

n es el grado del polinomio y U_j es la representación cartesiana de los polinomios de Zernike [1].

z_j^k son los coeficientes de la combinación lineal de los polinomios de Zernike y están denotados por:

$$z^k = [z_1^k, z_2^k, z_3^k, \dots, z_L^k]. \quad (8)$$

Donde k es un índice que corre sobre el número de elementos de la población del algoritmo DE y PSO que se encargan de optimizar el interferograma a demodular.

Por otra parte, el criterio de convergencia utilizado en el algoritmo está dado por:

$$\left| \frac{\alpha - f_{opt}(z^*)}{\alpha} \right| < \epsilon. \quad (9)$$

Donde z^* , representa el vector de la población con el mejor valor de optimización, ϵ es un error relativo de tolerancia. El criterio de paro que se definió es el cumplimiento del criterio de convergencia o se agotó el número de llamadas a función (NFL).

3. Descripción de los experimentos

Para el presente artículo, se realizaron cinco experimentos, generados a partir de un simulador de interferogramas, creado por los autores y basado en el modelo de la Ecuación (1). Cada experimento se generó utilizando la combinación lineal de polinomios de Zernike dado por la Ecuación (6), y los coeficiente z^k fueron generados a partir de valores aleatorios en el rango de $[-1,1]$. Los experimentos fueron capturados en imágenes de 40x40 pixeles, en la Fig. 1 se muestra un ejemplo de ello.

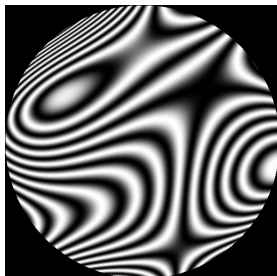


Figura 1: Interferograma modelado con polinomios de Zernike de cuarto grado.

4. Resultados experimentales

Para todos los experimentos se utilizó $\alpha = 5000$, $\lambda = 0$, $NLF = 7000$ y $\varepsilon = 0.0001$. En el caso ED se tomaron los siguientes parámetros para todos los experimentos $NP = 100$, $F = 0.5$ y $PC = 0.5$, por otra parte, para el algoritmo PSO se tomó $\omega = 1$, $\varphi_1 = 10$, $\varphi_2 = 8$ y $NP = 100$. El valor α es usual en la literatura y el resto de los valores fueron propuestos de manera experimental no exhaustiva.

Para cada experimento se realizaron 32 repeticiones con cada algoritmo y se muestran las demodulaciones que obtuvieron un menor error de ajuste.

4.1. Experimento 1

En la Fig. 2 (a) se muestra la imagen del interferograma del experimento 1, que fue generada con el siguiente polinomio de Zernike:

$$f(x, y) = 0.25U_2 + 0.15U_3 + 0.54U_4 + 0.12U_5 - 0.34U_6 + 0.25U_7 + 0.19U_8 - 0.26U_9 + 0.32U_{10} + 0.09U_{11} + 0.03U_{12} + 0.02U_{13} - 0.02U_{14} + 0.01U_{15}. \quad (10)$$

Donde $U_j = U_j(x, y)$ de la Ecuación (6), se usa por simplicidad.

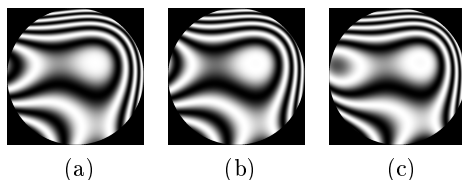


Figura2: El Interferograma original se muestra en (a), el interferograma recuperado por parte DE y PSO se muestra en (b) y (c) respectivamente para el experimento 1.

La Fig. 2 (b) muestra el resultado de recuperación por parte de DE alcanzando un error de ajuste de $\varepsilon = 0.019$, y en la Fig. 2 (c) muestra la recuperación para PSO alcanzado un error de ajuste $\varepsilon = 0.03$.

4.2. Experimento 2

En la Fig. 3 (a) se muestra la imagen de interferograma para el experimento 2, que fue generado con el siguiente polinomio de Zernike:

$$f(x, y) = 0.40U_1 + 0.97U_2 + 0.67U_3 + 0.15U_4 - 0.69U_5 + 0.23U_6 - 0.60U_7 + 0.26U_8 + 0.10U_9 - 0.15U_{10} - 0.39U_{11} - 0.17U_{12} + 0.64U_{13} - 0.71U_{15}. \quad (11)$$

En la Fig(s). 3 (b) y (c) se muestra el interferograma recuperado, con errores de ajuste del orden de $\varepsilon = 0.13$ y $\varepsilon = 0.11$ respectivamente .

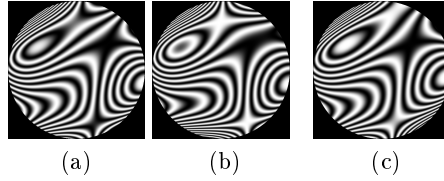


Figura 3: El Interferograma original se muestra en (a), el interferograma recuperado por parte DE y PSO se muestra en (b) y (c) repectivamente para el experimento 2.

4.3. Experimento 3

En la Fig. 4 (a) se muestra la imagen de interferograma para el experimento 3, que fue generado con el siguiente polinomio de Zernike:

$$f(x, y) = 0.1U_1 + 0.56U_2 + 0.34U_3 - 0.31U_4 - 0.41U_5 + 0.09U_6 + 0.78U_7 - 0.5U_8 + 0.12U_9 + 0.475U_{10} + 0.09U_{11} + 0.04U_{12} - 0.02U_{13} + 0.01U_{14} + 0.002U_{15}. \quad (12)$$

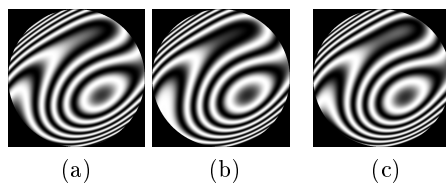


Figura 4: El Interferograma original se muestra en (a), el interferograma recuperado por parte DE y PSO se muestra en (b) y (c) repectivamente para el experimento 3.

En la Fig(s). 4 (b) y (c) se muestra el interferograma recuperado, con errores de ajuste del orden de $\varepsilon = 0.033$ y $\varepsilon = 0.039$.

4.4. Experimento 4

En la Fig. 5 (a) se muestra la imagen de interferograma para el experimento 4, que fue generado con el siguiente polinomio de Zernike:

$$f(x, y) = -0.40U_2 + 0.457U_3 + 0.939U_4 - 0.510U_5 + 0.262U_6 + 0.603U_7 + 0.947U_8 - 0.082U_9 + 0.580U_{10} - 0.181U_{11} + 0.381U_{12} - 0.023U_{13} + 0.270U_{14} - 0.136U_{15}. \quad (13)$$

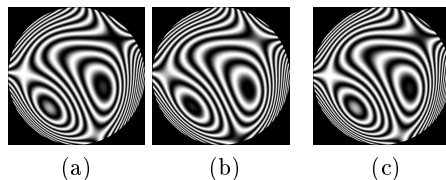


Figura 5: El Interferograma original se muestra en (a), el interferograma recuperado por parte DE y PSO se muestra en (b) y (c) repectivamente para el experimento 4.

En la Fig(s). 5 (b) y (c) se muestra el interferograma recuperado, con errores de ajuste del orden de $\varepsilon = 0.11$ y $\varepsilon = 0.06$.

4.5. Experimento 5

En la Fig. 6 (a) se muestra la imagen de interferograma para el experimento 5, que fue generado con el siguiente polinomio de Zernike:

$$f(x, y) = -0.86U_1 + 0.979U_2 - 0.09U_3 - 0.80U_4 - 0.03U_5 - 0.01U_6 + 0.09U_7 - 0.67U_8 + 0.05U_9 - 0.22U_{10} - 0.46U_{11} + 0.01U_{12} - 0.67U_{13} + 0.06U_{14} - 0.72U_{15}. \quad (14)$$

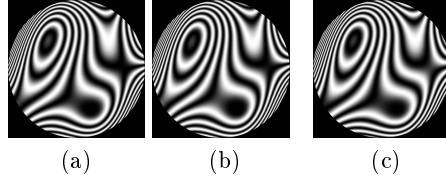


Figura 6: El Interferograma original se muestra en (a), el interferograma recuperado por parte DE y PSO se muestra en (b) y (c) respectivamente para el experimento 5.

En la Fig(s). 6 (b) y (c) se muestra el interferograma recuperado, con errores de ajuste del orden de $\varepsilon = 0.016$ y $\varepsilon = 0.047$.

4.6. Prueba de suma de rangos con signo de Wilcoxon

Para llevar a cabo la prueba se realizaron 32 repeticiones de cada experimento tomando los mismos parámetros iniciales para ambos algoritmos.

Los resultados estadísticos al aplicar los algoritmos DE y PSO en la demodulación de interferogramas, se presentan en la Tabla 1.

Tabla 1: Resultados estadísticos de la demodulación de interferogramas utilizando DE y PSO (los valores presentados en la tabla corresponden a $f_{opt}(z^*)$, para cada algoritmo).

Experimento	DE			PSO		
	Mediana	\bar{x}	σ^2	Mediana	\bar{x}	σ^2
1	4827.0	4820.9	3036.9	4694.0	4682.7	7456.8
2	4299.0	4265.3	2789.1	4421.0	4390.3	9605.4
3	4694.0	4708.9	4376.1	4648.0	4659.6	6771.4
4	4331.0	4346.7	2768.5	4475.0	4505.3	7899.1
5	4874.0	4867.29	1875.2	4699.0	4709.1	1779.3

Aplicando la prueba de suma de rangos con signo de Wilcoxon, Cuyos resultados se muestran en la Tabla 2, con los parámetros elegidos y las mismas condiciones para cada uno de los algoritmos, nos da como resultado que *no es posible determinar una diferencia entre las distribuciones de los dos algoritmos*.

Tabla 2: Resultados de la prueba de suma con signo de Wilcoxon.

T_+	T_-	T	T_0
9	6	6	1

Por otra parte, medimos los tiempos de ejecución para cada una de las repeticiones para ambos algoritmos, y obtuvimos que en promedio el algoritmo DE demora 925 segundos, mientras que PSO tarda 921 segundos, para una computadora con procesador AMD Turion(tm) 64 X2 Mobile Technology TL-56 \times 2, y 2GB en RAM.

5. Conclusiones

Los resultados muestran que bajo las condiciones mencionadas, el algoritmo DE no es mejor que PSO, para llevar a cabo la demodulación de interferogramas de franjas cerradas modeladas con polinomios de Zernike. Los parámetros iniciales utilizados para los algoritmos, fueron encontrados de manera experimental, quedando abierta la posibilidad de encontrar (proponer) otros parámetros que mejoren el desempeño de los algoritmos.

Cabe mencionar que DE tiene menor dispersión de resultados de ajuste de fase que PSO, basándonos en los resultados para la varianza presentada en la Tabla 1. Lo que nos permite concluir que DE muestra evidencia sobre una mayor eficiencia (reproducibilidad) de los resultados utilizando los mismos parámetros iniciales.

El presente artículo muestra evidencia de que es posible usar diferentes tipos de algoritmos de Soft computing, ya que se han presentado dos métodos que pertenecen a diferentes ramas de algoritmos de optimización, por su parte DE pertenece a los algoritmos Evolutivos, mientras que PSO a los algoritmos de optimización por cúmulo de partículas.

Agradecimientos. Agradecemos al Consejo Nacional de Ciencia y Tecnología (CONACYT), por el apoyo realizado para la elaboración del presente artículo y a la Dirección General de Educación Superior Tecnológica (DGEST, proyecto 4572. 12-P).

Referencias

1. Malacara, D.: Optical Shop Testing (Wiley Series in Pure and Applied Optics). Wiley-Interscience (2007)
2. Hariharan, P.: Basics of Interferometry. Elsevier Science (2010)
3. Born, M., Wolf, E., Bhatia, A.: Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light. Cambridge University Press (1999)

4. Takeda, M., Ina, H., Kobayashi, S.: Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry. *Journal of the Optical Society of America* (1917-1983) **72** (January 1982) 156
5. Womack, K.H.: Interferometric phase measurement using spatial synchronous detection. Volume 429 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. (January 1983) 8
6. Cuevas, F.J., Servin, M., Stavroudis, O.N., Rodriguez-Vera, R.: Multi-layer neural network applied to phase and depth recovery from fringe patterns. *Optics Communications* **181** (July 2000) 239–259
7. Cuevas, F., Sossa-Azuela, J., Servin, M.: A parametric method applied to phase recovery from a fringe pattern based on a genetic algorithm. *Optics Communications* **203** (2002) 213 – 223
8. Mancilla, L.E., Carpio, J.M., Cuevas, F.J.: Demodulation of interferograms of closed fringes by zernike polynomials using a technique of soft computing. *Engineering Letters* **15**(1) (2007) 99–104
9. Jiménez, J.F., Sossa, H., Cuevas, F.J.: Particle swarm optimization applied to interferogram demodulation. In: *EVOLVE*. (2012) 327–337
10. Storn, R., Price, K.: Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization* **11**(4) (December 1997) 341–359
11. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*. (1995) 1942–1948
12. Wilcoxon, F.: Individual Comparisons by Ranking Methods. *Biometrics Bulletin* **1**(6) (1945) 80–83
13. Carlisle, A., Dozier, G.: An off-the-shelf PSO. In: *PSO Workshop, Indianapolis, IN* (April 2001)
14. Wáng, Y., Lǐ, B., Weise, T., Wang, J., Yuan, B., Tian, Q.: Self-Adaptive Learning Based Particle Swarm Optimization. *Information Sciences – Informatics and Computer Science Intelligent Systems Applications: An International Journal* **181** (2011) 4515–4538
15. Bratton, D., Kennedy, J.: Defining a standard for particle swarm optimization. In: *IEEE Swarm Intelligence Symposium*. (2007) 120–127
16. Carpio, M., Servin, M., Malacara, D.: Direct phase detection of lateral shear interferograms using a phase-locked loop. *Optics Communications* **108** (June 1994) 225–229

Predicción de índices de reprobación escolar: un modelo de series de tiempo

Rosa-Irene Rojas-Rauda¹ y Omar-Jacobo Santos-Sánchez²

¹Instituto Tecnológico de Pachuca,
Pachuca, Hgo., México

²Universidad Autónoma del Estado de Hidalgo,
Pachuca, Hgo., México

Resumen En este artículo se presenta un modelo de predicción basado en la técnica Box-Jenkins, para estimar el índice de reprobación escolar del programa de la Licenciatura en Administración del Instituto Tecnológico de Pachuca (ITP). El modelo empleado es una función polinomial en el tiempo cuyos coeficientes son hallados mediante la bien conocida técnica de mínimos cuadrados. El orden de la función polinomial es propuesto y reajustado de acuerdo a resultados parciales de predicción estimados mediante la técnica. Los errores finales de predicción obtenidos, muestran la factibilidad del método.

Palabras clave: modelo Box-Jenkins, serie de tiempo, predicción, índices de reprobación.

1. Introducción

La técnica de predicción Box-Jenkins ha sido utilizada en las áreas de ingeniería y económico administrativas para proponer modelos dinámicos complejos o difíciles de modelar. Ejemplo de ello es el uso de esta técnica en la predicción de la demanda escolar en México hecha por el ANUIES [1]. En [2] la técnica de predicción Box Jenkins es empleada en el área de la econometría para predecir volúmenes de producción agropecuaria, precios de estabilización de productos, etc. En [3] se propone un pronóstico de matrículas para asignaturas mediante la técnica Box-Jenkins. Además de propone un desarrollo de software para el análisis y presentación de resultados. Otro tipo de técnicas han sido empleadas para predecir índices de reprobación, tal como la propuesta expuesta en [4], en la cual se emplea inteligencia artificial para predecir índices de reprobación en la Universidad Tecnológica de León. En [5] utilizando técnicas probabilísticas y análisis de datos se predicen las trayectorias de los alumnos en la Universidad Veracruzana. En el presente trabajo, se estudia la factibilidad del uso de un modelo matemático descrito mediante una función polinomial [6] para la predicción de índices de reprobación escolar en las materias de un programa educativo específico. El orden del polinomio es propuesto de acuerdo a un valor de error preestablecido. Para ello se cuentan con los registros de los índices de reprobación

de 12 semestres de la totalidad de las materias impartidas en la Licenciatura en Administración del Instituto Tecnológico de Pachuca. Para establecer el orden de la función polinomial de predicción y el umbral de error, se toman en cuenta solo 11 semestres, y se predice el último dato (ya conocido). Una vez establecido el orden de la función polinomial tal que satisfaga el umbral de error, se predijo el semestre Agosto-Diciembre del 2012 (éste es el treceavo semestre, el cual se conoció hasta diciembre del 2012). Los resultados obtenidos muestran la factibilidad del método propuesto. Con ésta predicción la coordinación de la carrera puede tomar acciones de planeación como programación de horarios, salones, número de grupos y profesores, realizar acciones correctivas tales como programación de cursos de capacitación y actualización docente. Cabe señalar que en el presente trabajo, solo un modelo sencillo de predicción fué propuesto, éste puede ser mas complejo y usar un modelo que involucre entradas externas (modelos ARMAX) las cuales representarán profesores específicos, factores socioeconómicos de los alumnos, condiciones generales en la institución, etc. El orden del artículo es el siguiente: en el capítulo 2 se presentan los resultados preliminares para realizar la predicción, en el capítulo 3 se presentan el análisis de los resultados obtenidos y finalmente en el capítulo 4 se presentan los comentarios concluyentes y el trabajo a futuro.

2. Resultados preliminares

En este capítulo se recuerdan los principales resultados técnicos útiles para resolver el problema de predicción planteado. Si se considera al proceso de cursar una materia por parte de un grupo específico, como una función que depende exclusivamente del tiempo, entonces es posible obtener un modelo como una función polinomial de la siguiente forma:

$$s(t) = d_0 + d_1 t + \dots + d_n t^n, \quad (1)$$

donde los coeficientes d_i , $i = 0, \dots, n$ son desconocidos y deben calcularse, $s(t)$ es el índice de reprobación a predecir. Este modelo es propuesto en [6]. El orden n es ajustado para satisfacer un umbral de error, el cual se calcula mediante una predicción parcial de resultados. Aunque este modelo es exclusivamente temporal, como se ve más adelante, es suficientemente útil como para predecir los índices de reprobación en forma aceptable. La ecuación (1) puede escribirse en forma matricial como:

$$s(t) = x^T(t)\Theta + e(t), \quad (2)$$

where $x^T(t) = [1 \ t \ \dots \ t^n]$, $\Theta = [d_0 \ d_1 \ \dots \ d_n]^T$ y $e(t)$ representa el error de estimación. Ahora sean los tiempos específicos t_1, t_2, \dots, t_m , los cuales representan los tiempos específicos en los que se obtuvieron los índices de reprobación en semestres anteriores. Así se obtienen los siguientes vectores:

$$\mathbf{S}(t) = \mathbf{X}(t)\Theta + \mathbf{e}(t), \quad (3)$$

donde

$$\mathbf{S}(t) = \begin{bmatrix} s(t_1) \\ \vdots \\ s(t_m) \end{bmatrix}, \mathbf{X}(t) = \begin{bmatrix} x^T(t_1) \\ \vdots \\ x^T(t_m) \end{bmatrix}, \mathbf{e}(t) = \begin{bmatrix} e(t_1) \\ \vdots \\ e(t_m) \end{bmatrix} \quad (4)$$

Defina el siguiente índice de error:

$$J = \mathbf{e}^T(t)\mathbf{e}(t), \quad (5)$$

y de (3) se tiene que

$$\mathbf{e}(t) = \mathbf{S}(t) - \mathbf{X}(t)\Theta \quad (6)$$

por lo que

$$\begin{aligned} J &= (\mathbf{S}(t) - \mathbf{X}(t)\Theta)^T (\mathbf{S}(t) - \mathbf{X}(t)\Theta) \\ &= \mathbf{S}^T(t)\mathbf{S}(t) - \Theta^T \mathbf{X}^T(t)\mathbf{S}(t) - \mathbf{S}^T(t)\mathbf{X}(t)\Theta + \Theta^T \mathbf{X}^T(t)\mathbf{X}(t)\Theta \end{aligned}$$

Para minimizar J con respecto al vector Θ , se deriva con respecto a Θ y se iguala a cero (condición necesaria para la optimalidad [6]) se obtiene la siguiente solución para el vector de parámetros Θ :

$$\Theta = (\mathbf{X}^T(t)\mathbf{X}(t))^{-1} \mathbf{X}^T(t)\mathbf{S}(t), \quad (7)$$

Si $\mathbf{X}^T(t)\mathbf{X}(t)$ es definida positiva se asegura que la solución exista y que haga que J alcance un mínimo [6]. Esta solución es el bien conocido algoritmo de mínimos cuadrados [6], brevemente recordada aquí.

3. Resultados obtenidos y discusión

El modelo dado por (1) y la solución (7) se usan en este artículo para predecir el valor del índice de desempeño en el tiempo. Esto se hizo realizando un programa sencillo en Matlab (Math works) para calcular los coeficientes del modelo bajo estudio, proponiendo un polinomio de orden 5. El orden de dicho polinomio fué propuesto heurísticamente. Observe que un grado relativamente alto en el polinomio de predicción (1), hace que el ajuste a los índices de reprobación anteriores sea muy preciso, sin embargo, en general se observó que esto hace tener errores de predicción muy altos. Se estudiaron 50 materias del programa educativo de Licenciatura en Administración, del Instituto Tecnológico de Pachuca, Pachuca. Hidalgo. De éstas 50 materias, se tienen datos de índices de reprobación a partir del semestre Agosto-Diciembre del 2006, sin embargo, es necesario señalar que 5 materias (Sociología Organizacional, Derecho Laboral, Seguridad Social, Entorno Macroeconómico de las Organizaciones y Economía Empresarial) desaparecieron del mapa curricular debido a que en el año del 2010 se puso en marcha un nuevo plan de estudios. Con respecto a éstas materias se calculó el índice de reprobación usando hasta el penúltimo dato disponible y validándolo con el último dato disponible. Desde el año de 2006 a junio del 2012, 825 grupos fueron aperturados, correspondientes a las 50 materias bajo estudio.

En promedio se tuvieron grupos de 25 personas (existieron grupos de hasta 45 personas como máximo y de 1 persona como mínimo). El dato obtenido mediante la solución (1, 7) se validó con el índice de reprobación obtenido en el semestre Agosto-Diciembre del 2012 y los resultados se muestran la Figura 1, los errores están dados en porcentaje y ordenados en forma ascendente. El listado de materias bajo estudio es el siguiente

- | | | | |
|---------------------|------------------------|----------------------|-----|
| 1. Admon/Calidad | 6. Derecho Mercantil | 11. Sem.Admón.PM.ME. | |
| 2. Competitividad | 7. Dis.Organizacional. | 12. Admon/Suel/Sala. | |
| 3. Form/Eval/Proye. | 8. Consultoría | 13. Tall/Inf/Adm II | (8) |
| 4. Admon/Finan.II | 9. Proc.Produc. | 14. Tall/Inves. I | |
| 5. Admon/Estrateg. | 10. Tall/ Inves. II | 15. Comunica Corp. | |

- | | | | |
|-----------------------|----------------------|---------------------------------|-----|
| 16. Des./Sustentable. | 21. Fund/Der. | 26. Función Admva.II | |
| 17. Prob.Soc.Mex. | 22. Ent/Macro/Org. | 27. Mat.Aplic.Admón | |
| 18. Soc/Organ | 23. Fundam.Econom | 28. Taller Ética | |
| 19. Seg/Social. | 24. Met/Cuant/Admon | 29. Estad. para Admón.II | |
| 20. Inform/p/Admón | 25. T.Gral.de Admón. | 30. Tall.Desarr.Hum. Organ.M.A. | (9) |

- | | | | |
|---------------------|----------------------|--|------|
| 31. Comport/Organ | 36. Econ. Internac. | | |
| 32. Fund/Invest. | 37. Costos Manuf. | | |
| 33. Sem/de Ética. | 38. Neg.Gest.Contra. | | (10) |
| 34. Dinámica social | 39. Diag/Eval/Empre | | |
| 35. Antropol.Filos. | 40. Función Admva.I | | |

- | | | | |
|------------------------|-------------------------|--|------|
| 41. Fund/Mercadotec. | 46. Der.Lab. Y Seg.Soc. | | |
| 42. Admón. Cap. Hum | 47. Adm/Cap/Hum II | | |
| 43. Conta.Gerencial. | 48. Mezcla/Mdo. | | (11) |
| 44. Comun/Mercadología | 49. Sis/Inf/Mdo | | |
| 45 Tall/Emprend. | 50. Econ.Empresarial. | | |

Como ya se mencionó durante 2010, la currícula de la carrera tuvo un rediseño, por lo que algunas materias cambiaron de nombre, sin embargo, en general el contenido temático se mantuvo, por lo que se consideraron los datos de la materia con el nombre anterior y el presente como una sola materia. En la lista dada se da el nombre actual de la materia. El menor error porcentual obtenido fué de 0% y el mayor fué de 20.52%. Esta diferencia, puede explicarse debido a que los índices de reprobación son afectados por cambios en los profesores que imparten las materias. En ese sentido, el error de predicción puede minimizarse aún más, mejorando el modelo. En efecto, un modelo que considere a los profesores como entradas externas puede mejorar los resultados obtenidos. adicionalmente pueden considerarse como perturbaciones factores socioeconómicos, malos hábitos de estudio, etc.

Remark 1. La correlación de los datos estaba bien condicionada debido a que en todos los casos la matriz $(\mathbf{X}^T(t)\mathbf{X}(t))$ estuvo bien condicionada y su inversa siempre existió. Esto da evidencia de la identificabilidad del proceso bajo estudio.

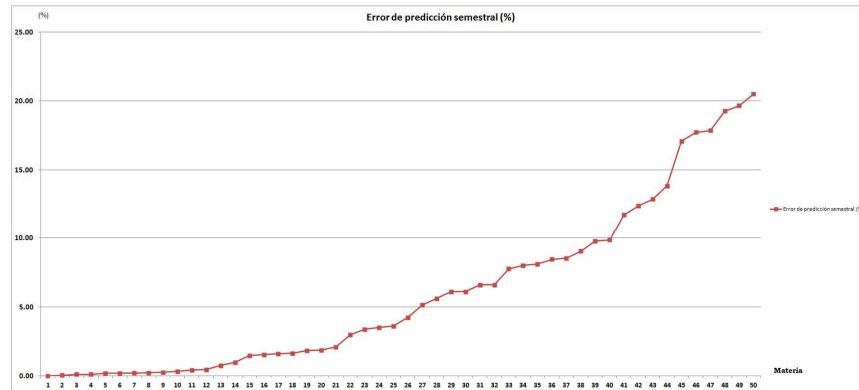


Figura 1. Resultados de predicción obtenidos.

Remark 2. El modelo propuesto (1) considera las siguientes suposiciones:

1. La totalidad (o la mayoría) de los profesores evalúan a los alumnos apegados al reglamento escolar.
2. El índice de deserción escolar es relativamente bajo.

4. Comentarios concluyentes

En el presente artículo se presentó un modelo polinomial temporal sencillo para la predicción de índices de reprobación de materias en una carrera específica. Los resultados de predicción obtenidos son útiles para una correcta planeación de requerimientos de espacios físicos y profesores. El modelo considera al fenómeno bajo estudio como una serie de tiempos. De acuerdo a los resultados obtenidos, el modelo de predicción brindó un buen desempeño. Dicho modelo puede mejorarse considerando un modelo ARX (AutoRegresive eXogenous) que considere dos tipos de factores externos: profesores y perturbaciones tales como factores socioeconómicos, entre otros, esto será objeto de trabajo a futuro.

Referencias

1. Kleiman, A. La previsin de la demanda de eduacin superior y los recursos necesarios para satisfacerla, Presentado a la XVI Asamblea General Ordinaria de la ANUIES celebrada en marzo de 1975, México.
2. Rosales Álvarez, R. A., et al. Fundamentos de econometría intermedia: Teoría y aplicaciones, Apuntes de clase, ISSN 1909-4442, enero 2010, Universidad de los Andes, Venezuela.
3. González, G., M. L. Pronstico de matrícula de asignaturas sin prerequisites mediante modelos basados en series de tiempo, Tesis de licenciatura, junio 2005, Caracas, Venezuela.
4. Reyes, A., N. G. Análisis de datos para la predicción del índice de reprobación en el área de tecnologías de la información y comunicaciones de la UTL, CITIBB 2012, México.
5. Chaín, R. R. Alumnos y trayectorias. Procesos de análisis de información para diagnóstico y predicción. ANUIES (ED.), deserción, rezago y eficiencia terminal en las IES. Propuesta metodológica para su estudio. Serie Investigaciones. 2001, México: ANUIES.
6. Wellstead, P.E. and Zarrop, M.B. Self tuning systemas: Control and signal processing, Jhon Wyley and Sons, 1991, England.

Modelado de sistemas adaptativos de inferencia neuro-difusa usando diseño central compuesto

Pascual Noradino Montes Dorantes¹, Rolando Praga Alejo¹,
Juan Pablo Nieto González¹ y Gerardo Maximiliano Méndez²

¹ Corporación Mexicana de Investigación en Materiales S.A. de C.V.,
Saltillo, Coahuila, México
pascual.montes@comimsa.com, cualmontes@hotmail.com,
rolandopraga@comimsa.com, juan.nieto@comimsa.com

² Posgrado en Ingeniería Mecatrónica, Instituto Tecnológico de Nuevo León,
Guadalupe Nuevo León, México
gmm_paper@yahoo.com

Resumen El diseño de experimentos, así como los sistemas de computación suave han sido ampliamente usados para evaluar el comportamiento de algún proceso. Este artículo presenta una metodología híbrida basada en ANFIS y un diseño central compuesto 2^k que sirve para la optimización de recursos evaluando únicamente algunas pruebas basándose en el método de diseño central compuesto, el cual proporcionara la base de datos necesaria para entrenar el ANFIS. Los resultados obtenidos muestran que el modelo es viable para la predicción de los parámetros necesarios para un proceso de maquinado y puede ser usado en un proceso real ahorrando recursos (humanos, tecnológicos, materiales, tiempo, entre otros) elaborando únicamente algunas pruebas para realizar los ajustes necesarios para mantener el control del proceso y los parámetros de calidad. El modelo propuesto también ayuda a reducir el número de reglas necesarias para la elaboración de sistemas del tipo neuro-difuso.

Palabras clave: ANFIS, modelo híbrido, diseño central compuesto, reducción de reglas, proceso de maquinado.

1. Introducción

Los procesos de maquinado han cobrado suma importancia en los procesos productivos hoy en día, estos procesos realizan el conformado de piezas por medio de corte, dichas piezas también son utilizadas en industrias que requieren de alta precisión (relojería, aeroespacial, militar y médica entre otras). Generalmente estos procesos de maquinado son delegados a personal con amplia experiencia y conocimiento. El diseño del proceso específico para una pieza depende de ciertos factores tales como los materiales y las herramientas necesarias para cortar dichos materiales, para lograr este proceso de manera satisfactoria es necesario entender el material y sus propiedades. El IJAT (International Journal of Automation

Technology) en su número especial llamado Maquinado: la dificultad para cortar materiales [1] menciona que las bases de datos, los manuales y las especificaciones no determinan con un modelo específico los parámetros necesarios para realizar el corte y se requiere de modelos que puedan simular las diferentes condiciones de trabajo para ahorrar costos; los modelos de computación suave permiten modelar estos procesos y por medio de simulaciones matemáticas evaluar dicho comportamiento sin la necesidad de tener un conocimiento del experto, aunque requerirán de tiempo de cómputo para poder entrenar al sistema y acercarlo a los parámetros óptimos, quedando algunas pruebas para hacer el ajuste fino de dichos parámetros sin la necesidad de desperdiciar grandes cantidades de material reduciendo la cantidad de pruebas a un mínimo.

Se han utilizado múltiples modelos para generar sistemas expertos, uno de los principales aspectos que hace que sea complicado modelar estos sistemas particularmente los modelos en donde interviene la lógica difusa, esto estriba en que se deben conocer las relaciones entre las variables para simplificar el modelo dejando fuera de este las variables que no estén correlacionadas, al eliminar estas variables se eliminan iteraciones y el número de reglas puede reducirse considerablemente, entre los modelos presentados se encuentran entre otros dentro del campo del diseño de experimentos los modelos de superficie de respuesta, presentados por Makadia y Nanavati [2] hacen una pequeña recopilación de los modelos utilizados para ajustar los parámetros en procesos de maquinado. Los modelos encontrados son: Metodología de Taguchi, Superficie de respuesta; por otro lado dentro se encontraron los modelos del diseño factorial presentados por Buragohain y Mahanta [3] y del diseño central compuesto presentado por Praga et al [4].

Martínez et al en [5], mencionan que por medio de los modelos de computación suave, particularmente la lógica difusa y las redes neuronales o modelos híbridos tipo ANFIS, se pueden tener dentro del sistema todos los posibles estados del mismo y no se requiere probar todos los estados, el comportamiento de dichos estados se puede aproximar por medio de un sistema difuso puro o por alguna técnica matemática de aproximación.

El caso de estudio de este artículo se basa en un proceso de maquinado expuesto por Praga et al en [4] y se ha adaptado para el modelo híbrido del diseño central compuesto con ANFIS. La sección 2 presenta un panorama general de la lógica difusa, la sección 3 muestra un panorama general del modelo ANFIS, en la sección 4 aparece el modelo del diseño central compuesto, la sección 5 describe el modelo híbrido y el caso de estudio, finalmente la sección 6 se muestran los resultados.

2. Lógica difusa

Este modelo de computación suave es una propuesta hecha por el Dr. Loftly Zadeh [6], Morales-Luna en [7] menciona que se puede definir como un modelo matemático para crear aproximaciones por medio de iteraciones vectoriales pudiéndose ajustar hoy en día como una extensión relajada de la lógica tradicional booleana; por otro lado Arredondo en [8] menciona que esta Lógica difusa (FL) utiliza conceptos de

pertenencia de sets (conjuntos) más parecidos a la manera de cómo piensa una persona.. Dentro de la lógica tradicional booleana, solo existen dos criterios de evaluación, el estado cierto o falso (Figura 2.1). En este caso no se tienen puntos intermedios de evaluación como se tienen en la lógica usada dentro de la vida real.

Esta lógica (FL) opera de la misma forma que el lenguaje, teniendo relaciones proposicionales que se convierten en reglas del modo (1). Estas reglas son combinadas de acuerdo a las entradas o valores asignados a las variables teniendo lo que se conoce como nivel de activación para cada regla (2) y posteriormente la función de base difusa (3) arrojará un valor difuso de salida que deberá transformarse a un escalar o a un valor lingüístico para obtener el resultado de la evaluación.

$$\text{Si } x_1 \text{ y } x_2 \text{ y } \dots \text{ y } x_n \text{ entonces } z \quad (1)$$



Figura 2.1 Esquema del espacio de pertenencia de la lógica Booleana

$$\mu R \circ Sx, = \sup_{y \in V} [\mu R(x, y) \star \mu S(y, z)] \quad (2)$$

donde \star denota el producto

$$FBF = \frac{\sum_{i=1}^n A^i_i}{\sum_{i=1}^n \prod A^i_i} \quad (3)$$

3. Modelo ANFIS

Este tipo de sistemas fue propuesto por primera vez en el año de 1993 por Jang en [9]; Kabini et al. en [10] mencionan que se unió lo mejor de los dos modelos (Redes Neuronales y Lógica Difusa tipo 1) para crear un sistema híbrido que fuera capaz de adaptarse por medio de retroalimentación.

Básicamente el modelo original propuesto por Jang en 1993 [9] no ha tenido una evolución significativa, se continua usando el mismo bajo el concepto propuesto, Jang et al en [11], el cual consiste de 5 capas básicas; existen algunos trabajos que presentan combinaciones híbridas de diferentes modelos heurísticos o estocásticos con la arquitectura del sistema ANFIS, aunque se encuentran estas aportaciones la base del funcionamiento del ANFIS es la misma, los trabajos presentados hacen aportes en el área del entrenamiento sin embargo la formulación matemática sigue siendo en 5 capas (Fig. 3.1). Entre los trabajos encontrados, se han reportado entre otros modelos los siguientes:

- Jang et al en [9] presentaron la Retro propagación.
- Hosseini et al utilizaron: mínimos cuadrados en [12]

- González y Ballarín usan: Transformaciones morfológicas (Transformada de Watershed). [13]
- Dutra y Martucci utilizan: Árboles de decisión en [14]

Un sistema neuro-difuso adaptativo (ANFIS, por sus siglas en inglés) es equivalente a un sistema difuso Sugeno (Fig. 3.2), el sistema Sugeno opera mapeando los valores de entrada en conjuntos difusos, obteniendo así resultados porcentuales de pertenencia que darán un valor generado por la combinatoria como salida del sistema, resultando una función la cual es evaluada por medio del sistema difuso. Por otra parte un ANFIS opera mapeando los valores porcentuales arrojados por un fuzzificador obteniendo como resultado un vector donde se establecen todos los valores ponderados (pesos) de las reglas activadas dando finalmente un valor por medio de una sumatoria teniendo como resultado una aproximación equivalente a un sistema difuso puro.

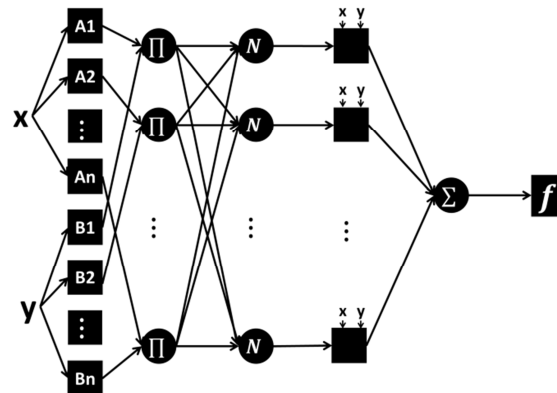


Figura 3.1. Representación gráfica de un sistema ANFIS.

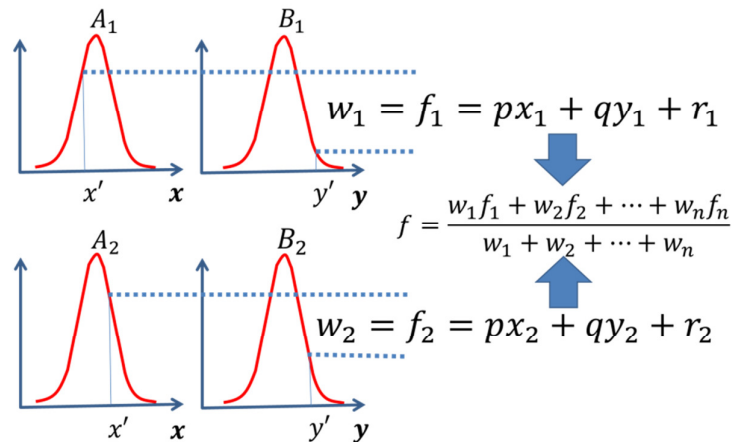


Figura 3.2. Representación gráfica del funcionamiento de un sistema difuso Sugeno

La arquitectura del sistema trata de simplificar las formulaciones matemáticas necesarias para la elaboración de un sistema difuso completo aunque (elimina el módulo de predicción o aproximación), usa un conjunto de reglas y un fusificador, el sistema híbrido ANFIS arroja un valor de salida entero sin la necesidad de que sea defusificado.

Este modelo se compone de 5 capas de neuronas (Fig. 3.1), cada una de ellas está dedicada a elaborar una función en particular. Debido a la simplicidad de las neuronas artificiales estas pueden realizar manipulación matemática a modo de operaciones simples y al estar restringidas se requieren de múltiples capas de neuronas para poder lograr una función compleja, algunos autores mencionan que se crearon estos tipos de modelos para no lidiar con el modelado matemático de un sistema difuso, pero aún se requiere establecer las etiquetas para los conjuntos difusos, además de establecer sus límites y la forma de sus funciones de membresía.

La capa 1 obtiene los valores de las funciones de membresía para las entradas. En un sistema difuso común generalmente se activan un par de reglas (4) para cada valor, el número de reglas activadas dependerá de la separación de los conjuntos. Se puede dar el caso de que solo se active una regla cuando el valor de prueba este centrado en la media. La activación de reglas se realiza de la siguiente manera:

$$\text{Regla } n: \text{ si } X_i \text{ es } A_i \text{ y } Y_i \text{ es } B_i, \text{ entonces } f(x,y) = pxi + qi yi + ri \quad (4)$$

Donde p, q y r son los coeficientes de las variables de la regla difusa.

La salida de esta capa arrojará (5, 6) teniendo así los valores de activación de cada variable en cada regla.

$$O_{1,i} = \mu_A(x), \quad \text{para } i = 1, 2, \dots, n \quad (5)$$

$$O_{1,i} = \mu_{B_{i-n}}(y), \quad \text{para } i = (n + 1), (n + 2), \dots, m \quad (6)$$

Donde los parámetros de $\mu_A(x)$, $\mu_B(y)$ se obtendrán de (7).

$$\text{gaussiana } (x; c, \sigma) = e^{-\frac{1}{2} \left(\frac{x-c}{\sigma} \right)^2} \quad (7)$$

La capa 2 arrojará como resultado el valor w_i . que indica el grado de activación para cada regla por medio del producto de la activación de las variables (8). En esta capa los nodos son fijos.

$$O_{2,i} = w_i = \mu_A(x) \cdot \mu_B(y), \quad i = 1, 2 \quad (8)$$

De esta capa se puede obtener lo que se puede denominar el valor fusificado de salida de la regla activada.

La capa 3 generará un vector donde se agrupan los valores de salida de cada estado en el punto i (9)

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \quad (9)$$

La **capa 4** generará los valores de salida de la función equivalente a la salida fusificada de cada regla activada (10) en el sistema Sugeno. (Fig. 3.1)

$$O_{4,i} = \bar{w}_i f = \bar{w}_i (p_i x + q_i y + r_i) \quad (10)$$

La **capa 5** genera el valor real del sujeto mediante la función (11).

$$O_{5,i} = \sum \bar{w}_i f_i = \frac{\sum w_i f_i}{\sum_i w_i} \quad (11)$$

4. Diseño central compuesto

Este tipo de diseño ha sido usado cuando se trata de estudiar dos o más factores relacionados para evaluar sus resultados. Como uno de los modelos más importantes se encuentra el de N^k donde N representa el número de niveles y k el número de factores a evaluar. Para el caso particular de estudio se seleccionó un modelo 2^k con puntos centrales y axiales (Fig. 4.1), teniendo los niveles máximo y mínimo como factores de los parámetros de entrada. Montgomery en [15] menciona que este modelo es de gran utilidad durante las etapas iniciales de los trabajos experimentales puesto que proporciona un número pequeño de corridas necesarias para evaluar el comportamiento dentro de un intervalo de valores, sin embargo en este caso en particular se ha usado para reducir la cantidad de reglas presentes en el modelo del sistema difuso.

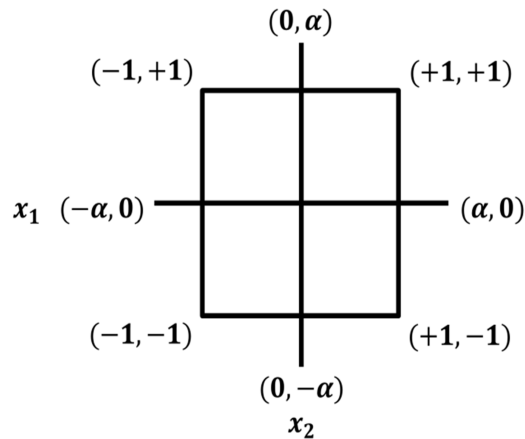


Figura 4.1 Diseño central compuesto, adaptado de Praga et al [4]

5. Modelo híbrido

5.1. Metodología

- A. Modelar un sistema MISO (Entradas Múltiples y una salida) basado en un sistema difuso puro que servirá de base para la evaluación así como para el ensamblado del ANFIS.
- B. Generar un espacio de partición uniforme abriendo los puntos axiales marcados por Praga en [4].
- C. Aproximar los resultados faltantes en la base de datos por medio de un modelo de mínimos cuadrados a partir de los datos de A.
- D. Obtener el modelo de reducción de reglas por medio de un diseño central compuesto con los datos obtenidos en C, generando la base de inferencia y los valores para cada variable en el estado -1,1 para las entradas y las salidas del modelo ANFIS optimizado.
- E. Generar la salida del modelo con los consecuentes usados en las reglas del sistema difuso y del sistema ANFIS teniendo aquí el modelo híbrido (ANFIS/2^k) optimizado del algoritmo.

5.2. Aplicación ANFIS

Como una propuesta novedosa que optimiza el modelo presentado por Praga-Alejo [4] se elaboró esta metodología, el método desarrollado mejora desempeño del modelo propuesto por [4], el cual consta de 11 reglas mientras el modelo mejorado (ANFIS/DCC) solo opera con 4 reglas obtenidas de los puntos axiales del DCC y optimiza el modelado del sistema reduciéndolo en 60% aproximadamente, además mejora la aproximación reduciendo la tasa de error, lo cual fue comprobado por medio del desarrollo de la aplicación de ajuste de parámetros de maquinado.

A partir de la tabla 5.1 se obtiene un sistema difuso puro T1SFLS (Sistema Difuso Singleton del tipo 1) que servirá de base para la evaluación así como para el ensamblado del ANFIS, generando un espacio de partición uniforme abriendo los puntos axiales. La Fig. 5.1 presenta la partición del universo de propagación de la variable X1(RPM), mientras la Fig. 5.2 presenta la partición del universo de propagación de X2 (Velocidad de avance), (Tabla 5.2), los resultados faltantes se aproximaron por medio de tabla 5.1 obteniendo la base de datos para el diseño central compuesto que realizará la reducción de reglas a partir de los datos obtenidos en C, generando la base de inferencia (tabla 4.3) y los valores para cada variable en el estado -1,1 para las entradas y las salidas del modelo ANFIS optimizado, La salida del modelo se generará con un modelo híbrido (ANFIS/2^k) optimizado.

Tabla 5.1 Diseño central compuesto, adaptado de: Praga et al, [4]

Entradas		Salida
X1	X2	Diámetro (mm)
2500.00	300.000	4.18

Entradas		Salida
X1	X2	Diámetro (mm)
2750.00	275.000	4.08
3000.00	250.000	4.19
2750.00	239.645	4.07
2396.45	275.000	4.15
2750.00	310.355	4.10
2750.00	275.000	4.12
2500.00	250.000	4.17
3103.55	275.000	4.08
3000.00	300.000	3.97
2750.00	275.000	4.03

Tabla 5.2 Base de reglas para las aplicaciones FLS y ANFIS (Antecedentes).

X1	X2
2250	225
2250	250
2250	275
2250	300
2250	325
2500	225
2500	250
2500	275
2500	300
2500	325
2750	225
2750	250
2750	275
2750	300
2750	325
3000	225
3000	250
3000	275
3000	300
3000	325
3250	225
3250	250
3250	275

X1	X2
3250	300
3250	325

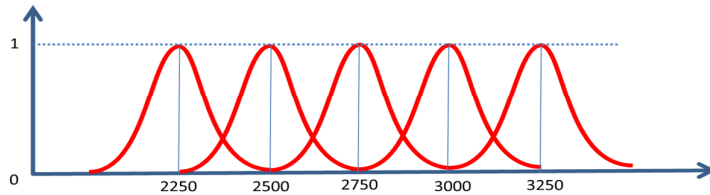


Figura 5.1 Universo de propagación de X1, adaptado de Praga et al [4]

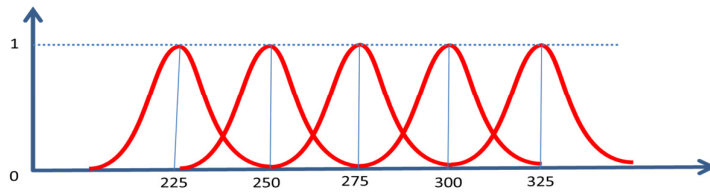


Figura 5.2 Universo de propagación de X2, adaptado de Praga et al [4]

Tabla 5.3 Base de reglas para la aplicaciones ANFIS/ 2^k.

X1	X2	Y
2250	225	4.1575
2250	325	4.0875
3250	225	4.0575
3250	325	3.9875

La base de datos (tabla 5.2) que fue tomada para modelar el sistema difuso de referencia servirá de base para la aplicación ANFIS, que posteriormente se optimizará. El sistema difuso creado utiliza: un modelo tipo Mamdani (T1SFLS), con un fusificador gaussiano y una implicación de producto, con un defusificador de centro promedio y 25 reglas para el modelo base. El modelo optimizado tiene las mismas características pero únicamente utiliza 4 reglas difusas generadas por el modelo del diseño central compuesto.

Las aproximaciones de los diferentes modelos incluyendo el híbrido desarrollado se describen en la figura 5.3

6. Resultados

Los resultados obtenidos de los diferentes experimentos muestran que el modelo híbrido generado es viable para su uso con el beneficio que solo requiere del establecimiento de cuatro reglas derivadas de los puntos axiales del diseño central compuesto. Las etiquetas para el establecimiento de la partición del universo de

propagación de los datos se pueden obtener por medio de la metodología Macvicar-whelan [16], el modelo híbrido ofrece la ventaja de operar con muy pocas reglas brindando resultados con una precisión aceptable basada en el diseño central compuesto con el requerimiento de elaborar únicamente 4 pruebas y poder aproximar los resultados dentro de todos los estados del sistema por medio del modelo ANFIS/2^k.

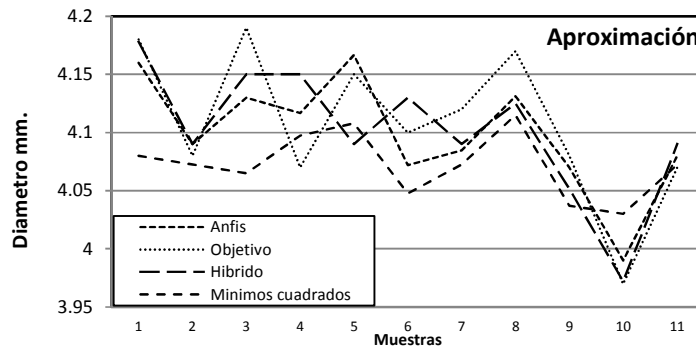


Figura 5.3 Resultados de la aproximación

La tabla 6.1 muestra la tasa de error de la aproximación obtenido con el indicador del error cuadrado medio (12) demostrando que la metodología híbrida del tipo Anfis/2^k es equivalente a un sistema difuso puro (25 reglas, 80% mayor en tamaño que el modelo propuesto) y ofrece mejores resultados que un modelo de regresión (mínimos cuadrados).

Tabla 6.1 Error promedio del modelo

Error cuadrado medio	Anfis	FIs	Anfis/DCC	Mínimos cuadrados
	0.00097	0.00076304	0.0015219	0.003786717

$$ECM = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{2} \quad (12)$$

En base al comportamiento observado se planea la realización de un modelo del tipo no-singleton como trabajo futuro; este modelo toma en cuenta la incertidumbre en los datos, esta puede presentarse entre otras formas como algún tipo de ruido procedente de la medición o del propio modelo tratando de generar aproximaciones más exactas.

Referencias

1. Fujipress. CALL FOR PAPERS, Special Issue on Machining of Difficult-to-cut Materials for the International Journal of Automation Technology (IJAT) Vol.7 No.3, May 2013

- [Internet] [Consultado el 28 de Marzo de 2013]. Disponible en: http://www.fujipress.jp/IJAT/CFP/ijat7-3_cfp.pdf (2012)
2. Makadia A.J. y Nanavati J. I.: Optimisation of machining parameters for turning operations based on response surface methodology. *Measurement* 46 (2013) pp.1521–1529 (2013)
 3. Buragohain M. y Mahanta C.: A novel approach for ANFIS modelling based on full factorial design. *Applied Soft Computing* (8) pp. 609–625 (2008)
 4. Praga-Alejo R., González G. D., Pérez V. P, Cantú S. M., Flores H. B.: Modeling a Fuzzy Logic System Using Central Composite Design. *Proceedings of 1st annual world Conference of the Society for Industrial and Systems Engineering*. Washington D.C., USA, Septiembre 16-18 (2012)
 5. Martínez, R., Castillo, O., Aguilar, L.T.: Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. *Information Sciences*. Vol. 179(13), pp2158-2174 (2009)
 6. Zadeh L.A.: Fuzzy Sets. *Information and control* Vol. 8, pp.338-353 (1965)
 7. Morales-Luna G.: Introducción a la lógica difusa, Centro de Investigación y Estudios Avanzados del IPN, (CINVESTAV-IPN), [Internet][Consultado el 8 de Octubre de 2011]. Disponible en: <http://delta.cs.cinvestav.mx/~gmorales/ldiffl/ldiffl.p> (2002)
 8. Arredondo T.: Introducción a la Lógica Difusa, [Internet] [Consultado el 2 de Septiembre de 2011]. Disponible en:<http://profesores.elo.utfsm.cl/~tarredondo/info/soft-comp/Introduccion%20a%20la%20Logica%20Difusa.pdf> (2009)
 9. Jang J.R.S.: ANFIS:Adaptative-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3), pp. 665-685 (1993)
 10. Kabini S.K., Ikua B.W. and Nyakoe G.N.: Review of ANFIS and its Application in Control of Machining Processes. *2011 Sustainable Research and Innovation Conference Proceedings* 3 (2011)
 11. Jang R.S., Sun C, Mizutani E.: *Neuro-Fuzzy and Soft Computing. A computational approach to learning and machine intelligence*. pp. 95-168 Upper Saddle River, NJ: Prentice-Hall (1997)
 12. Hosseini Monireh Sheikh, Zekri Maryam: Review of Medical Image Classification using the Adaptive Neuro-Fuzzy Inference System. *Journal of Medical Signals & Sensors*. 2(1). pp. 49-60 (2012)
 13. González M. A. y Ballarin V. L.: Segmentación de imágenes utilizando la transformada Watershed: obtención de marcadores mediante lógica difusa *IEEE LATIN AMERICA TRANSACTIONS* 6(2), pp. 223-228 (2008)
 14. Dutra R. G. y Martucci Jr. M.: Adaptive Fuzzy Neural Tree Network *IEEE LATIN AMERICA TRANSACTIONS*, 6(5), pp. 456-460 (2008)
 15. Montgomery D.C.: *Diseño y Análisis de experimentos*. 2ª Edición. Limusa-Wiley. Pp.218-279 (2004)
 16. Macvicar-Whelan P. J.: Fuzzy Sets, the Concept of Height, and the Hedge VERY. *IEEE transactions on systems, man, and cybernetics*, vol. Smc-8, no. 6, june 1978, pp 507-511 (1978)

Sistema de predicción de patrones de juego

Aram B. González y Jorge A. Ramírez Uresti

Departamento de Tecnologías de Información y Computación,
Campus Estado de México del Tecnológico de Monterrey,
Atizapán, Estado de México, México
{a00965707, juresti}@itesm.mx

Resumen Los sistemas multiagentes permiten representar ambientes en los que se requiere la interacción y cooperación de un conjunto de individuos. Mediante el modelado del oponente, se logra producir un concepto de los deseos del rival en el caso de ambientes competitivos. Se presenta el Sistema de Predicción de Patrones de Juego (SPPJ) que permite la predicción del movimiento del balón en posesión del rival basado en un análisis de los juegos previos de los equipos.

Palabras clave: sistemas multiagentes, razonamiento basado en casos, modelado del oponente.

1. Introducción

Se pueden definir a los agentes inteligentes como un sistema computacional que se encuentra en algún ambiente y que es capaz de llevar a cabo acciones autónomas dentro del mismo con el fin de lograr el objetivo para el que fue diseñado como indica [1]. Sin embargo, no existe una definición que abarque todas las características que un agente puede contener ya que pueden diferir en ciertas propiedades dependiendo del problema que atacan y de la implementación específica de su solución.

Una propiedad fundamental en el desarrollo de los sistemas que incorporan agentes es la interacción de los mismos con el medio en que se encuentran debido a que se espera que a cada acción realizada produzca una reacción en el mismo o una salida del sistema. Los agentes, en la mayoría de los casos, no tienen una vista global de su ambiente y tampoco ejercen un control total sobre él. Debido a las características de los sistemas multiagentes, uno de los campos de estudios en que son utilizados es la representación de ambientes competitivos en los cuales se enfrentan dos equipos. Un ejemplo es el uso del fútbol como ambiente. El objetivo fundamental del juego consiste en realizar más anotaciones que el oponente dentro de un campo de juego en base a una serie de reglas predefinidas.

En los ambientes competitivos, los equipos buscan maneras de lograr facilitar la realización de su objetivo primario. Cada equipo puede definir su manera de jugar, puede hacerlo en base a todos los elementos con los que cuenta o en base a un subconjunto de ellos. Los comportamientos derivados de este tipo de procesos

frecuentemente se pueden generalizar en patrones y jugadas que distintivamente se llevan a cabo a lo largo de los juegos.

Una de las mejores maneras para conseguir una ventaja sobre el rival, se basa en conocer de antemano las posibles opciones que frecuentemente utiliza para sobrellevar una situación en particular. Al conocer las acciones de un equipo, se puede determinar un plan que contrarreste sus medidas.

El resultado de predecir el comportamiento y los movimientos basados en un modelo en particular se le denomina Modelado del Oponente (MO). MO no especifica una técnica de aprendizaje automático en particular para llevar a cabo la creación del modelo por lo que los algoritmos y métodos quedan a criterio del investigador y normalmente se elige uno que se adapte a las condiciones del ambiente y del objetivo que se busca realizar.

MO puede ser creado en diversos ambientes aunque el modelo generado tiende a estar fuertemente relacionado con el entorno en el que se desarrolló debido a que generalmente se basa en los objetivos que se buscan conseguir.

En general, para crear un modelo del oponente se emplean técnicas de aprendizaje automático por lo que la cantidad de patrones puede variar pero la tendencia es partir de un número considerable de ejemplos. Se pueden emplear heurísticas para conseguir una base de entrenamiento que permita generalizar el modelo sin ocasionar un sobre-aprendizaje.

Varios estudios hacen uso de MO [2,3,4] para mostrar una mejoría entre un plan de juego básico y el que incorpora esta técnica, mientras que también se llega a su uso en la competencia de Coach de RoboCup [5]. Sin embargo, debido a la cantidad de datos requerida, en muchas pruebas se tiende a emplear un subdominio para llevar a cabo los experimentos. Mientras que los subdominios permiten llevar un mayor control sobre los factores que alteran el ambiente, también se entiende que los modelos generados pueden no llegar a ser suficientes para dominios que abarquen más acciones.

El MO debe ser al menos capaz de proveer información para que los agentes o miembros de un equipo, puedan anticipar la mayoría de los movimientos del oponente con cierta precisión. Esto debe conservarse inclusive cuando se enfrentan a oponentes que no habían sido presentados o estudiados previamente.

Han existido experimentos en los que se ha intentado mostrar la importancia de crear un buen modelo. Parker et al. [6] decidieron probar dos casos extremos, el primero consistió en actuar como si el oponente fuese a escoger siempre la mejor opción dada una situación. El segundo caso consistió en no tomar en cuenta el conocimiento que se tiene sobre el oponente. Esto fue probado bajo un juego de información incompleta y debido a esto, la suposición de que un oponente pueda tomar siempre la mejor opción es incorrecta. Los resultados generales mostraron este error ya que las pruebas en las que no se tomaba en cuenta al oponente, obtuvieron mejores resultados que las otras. La causa principal que fue atribuida es que reaccionar a la mejor opción posible en este tipo de ambientes es inútil ya que el oponente no conoce todos los factores que la rodean y por lo tanto difícilmente llegará a emplear dicha opción.

La creación de un modelo del oponente en tiempo real es uno de los principales retos existentes dentro del área. Las dos mayores complicaciones que se presentan se encuentran ligadas entre sí. La primera consiste en la cantidad de datos requeridos para crear el modelo, esto se traduce en tener que considerar un número considerable de interacciones que permitan generalizar el modelo. En el caso de un juego de fútbol, la cantidad de interacciones entre jugadores dentro de un juego no es suficiente para poder deducir la estrategia general del equipo [7]. La segunda complicación radica en el tiempo requerido para procesar los datos obtenidos y se deriva de la cantidad de información acumulada. Esto obliga a que se deba tener un conocimiento previo del rival en lugar de comenzar a armar el modelo en tiempo de juego.

Dado que siempre se debe saber reaccionar ante los rivales sin importar si fueron estudiados con anterioridad o si se desconocen por completo, se han hecho recomendaciones al respecto. Una de las posibilidades consiste en tener un plan de contingencia [8] que consiste en tener un modelo lo más genérico posible que abarque solamente información básica. La idea consiste en que esto se emplee como un último recurso cuando no se logra determinar el tipo de rival o simplemente no encaja en alguno de los modelos con los que se cuenta.

En el caso de esta investigación, se decidió que el ambiente a utilizar para el Sistema de Predicción de Patrones de Juego (SPPJ) sería el Simulador 2D de RoboCup. La decisión se basa en que se presenta un ambiente dinámico, con reglas estructuradas, basado en un deporte que es popular en la mayor parte del mundo, se tienen registros de competencias pasadas, se cuenta con los ejecutables para los equipos que ingresan a la competencia y el entorno permite ser modificado si es requerido en su caso [9].

En el fútbol, las estrategias se basan en la posición y posesión del balón debido a que es el factor más determinante en el campo puesto que el objetivo del juego gira a su alrededor. El poder determinar la ruta que el balón seguirá en posesión del rival permite a un equipo reaccionar de modo que pueda detener al oponente en su camino a lograr su objetivo. Las rutas del balón crean así un modelo indirecto del oponente ya que gracias a ellas podemos conocer los patrones de jugadas que emplearán sin tener que considerar todos los movimientos de los jugadores (sólo a los que participan o tienen alguna interacción con el balón). El seguir la ruta del balón también nos muestra los puntos en los que los jugadores deben estar colocados para realizar las jugadas.

Esta investigación está basada en las acciones defensivas de un equipo dentro de un sistema multiagentes que simula un juego de fútbol. La estructura del artículo está organizada de la siguiente forma. La sección 2 comprende creación de la base de conocimiento incluyendo la recolección y limpieza de datos. Las características dentro del desarrollo de un juego son discutidas en la sección 3. La sección 4 contiene los resultados obtenidos empleando el SPPJ. Por último, la sección 5 aporta una serie de conclusiones y trabajo a futuro contemplado.

2. Creación de la base de conocimiento

Para crear un modelo del oponente útil para el SPPJ, es necesario tomar en cuenta una serie de experiencias previas. Dado el tipo de ambiente y la accesibilidad de las fuentes, se emplearon registros de juegos pasados. Los registros son generados automáticamente por el Simulador 2D de RoboCup cada que un juego es ejecutado y se guarda en un archivo de tipo RCG.

Los archivos RCG son archivos binarios que pueden ser transformados en archivos del tipo XML en los que se guarda toda la información correspondiente a al juego incluyendo los parámetros del servidor, parámetros de los jugadores, estado del juego y las acciones de cada jugador.

Debido a que estamos empleando un modelo del oponente basado en la posición del balón, no toda la información contenida nos es útil por lo que decidimos eliminarla. Este proceso reduce el tamaño de los archivos y resulta en un XML que contiene las acciones de los jugadores, las posiciones de los jugadores, la posición del balón y el estado del juego.

Por cuestiones del tiempo que le toma a los jugadores desplazarse físicamente dentro del terreno de juego y tomando en cuenta el tiempo requerido para procesar las predicciones, se ha decidido que los patrones incluidos dentro de la base de conocimientos deben abarcar al menos 10 pasos en tiempo del servidor. No consideramos otro tipo de restricción al momento.

Para reducir la complejidad del modelo del oponente, se decidió evitar el uso de posiciones absolutas y en su lugar usar zonas que cubren cierto espacio. La división también permite al SPPJ ser tolerante al ruido provocado por el entorno, en este caso el Simulador 2D de RoboCup.

La división en zonas ha sido empleada con anterioridad [10,11], sin embargo, no se menciona una metodología específica que permita la obtención de un tamaño ideal por zona. En general, la división en zonas en este tipo de ambientes se lleva a cabo de manera empírica.

En nuestro caso, la división debía cumplir con que cada región debe ser lo suficientemente grande para reducir la complejidad pero también debe ser suficientemente pequeña para que la predicción continúe siendo relevante. La razón se encuentra en que tener una división muy pequeña daría demasiadas combinaciones para el momento de la búsqueda, desapareciendo así la ventaja de crear una división. Por el contrario, tener zonas de tamaño demasiado grande, nos daría un espacio de búsqueda pequeño pero la precisión en cuanto al lugar en el que el balón se encuentra sería irrelevante para los defensores puesto que abarcaría demasiado espacio.

El resultado de la división fue la creación de una malla de 60 zonas. La división también tiene la ventaja de tomar como puntos las banderas existentes dentro del campo del Simulador 2D de RoboCup. La Figura 1 muestra la división obtenida.

Se ha optado por generar un árbol de búsqueda que es empleado en la fase de juego y es el enlace entre el juego actual y las experiencias previas. El árbol permite reducir el tiempo empleado para comparar la situación actual del juego contra los patrones establecidos. La estructura del árbol está compuesta por una

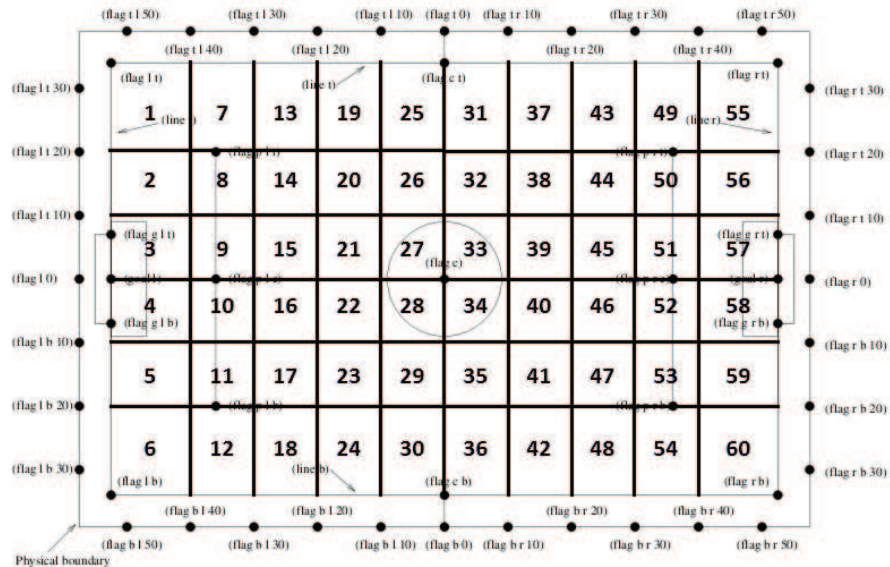


Figura 1. División del campo empleada en el SPPJ.

serie de zonas de interés que indican la colocación inicial tanto del balón como de los jugadores rivales al inicio de una jugada. Como muestra la Figura 2, cada rama del árbol de búsqueda corresponde a un patrón que puede presentarse dentro del juego, empezando por la zona del balón y siguiendo por las zonas en las que se encuentran los jugadores.

3. Fase de juego

Para hacer uso de la base de conocimiento creada en la fase anterior, el SPPJ fue probado en nuevos juegos ante varios rivales. Dentro de esta etapa se debe seguir el proceso completo: obtener la información actual del juego (posiciones del balón y de los rivales), búsqueda de patrones similares dentro de patrones similares dentro de la base de conocimientos (a través del árbol de búsqueda) y obtención de las posibles zonas en las que el balón se encontrará en determinado tiempo.

Para obtener el estado actual del juego se debe obtener la información parcial que cada agente conoce. Sin embargo, los parámetros por defecto del Simulador 2D de RoboCup limitan la comunicación a que cada agente puede escuchar un mensaje aleatorio por ciclo. Debido al enfoque del SPPJ, la forma en cómo se debe lidiar con una comunicación limitada queda fuera del alcance por lo que para efectos de las pruebas, se decidió eliminar dicha limitación.

Aún con los parámetros de comunicación modificados, se debía elegir a quien debería llevar a cabo el proceso de predicción. Los enfoques pensados fueron una decisión distribuida, que cada agente haga el proceso completo de predicción o

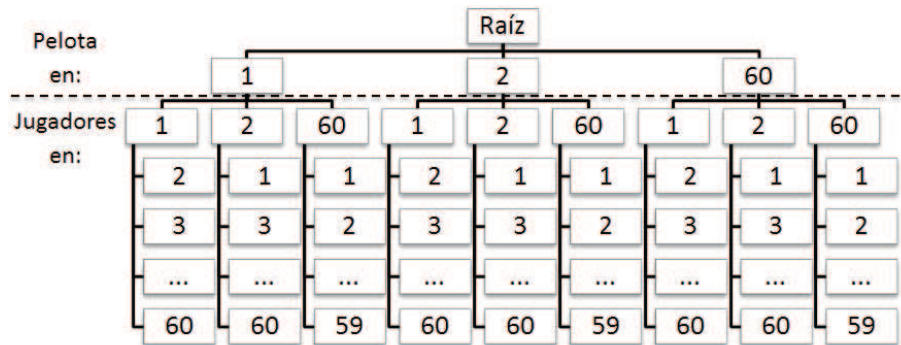


Figura 2. Estructura del árbol de búsqueda.

que un solo agente obtenga la información, complete el proceso y comunique los resultados. Las primeras dos opciones implicaban una gran cantidad de mensajes dentro del servidor y principalmente, los jugadores debían distraerse de su labor en el juego para llevar a cabo el proceso por lo que dejaban pasar sus acciones básicas por estar procesando. Debido a esto, se decidió tener un punto central que llevara a cabo el proceso y que sólo requiriese que la información de los agentes le fuera enviada. Para evitar otra negociación completa para decidir quién sería el agente encargado, se decidió optar por el portero debido a que es el jugador que tiene la menor cantidad de interacciones con el balón y que tiene la mayor visión del campo de juego.

Una vez que el portero recibe toda la información, debe ser limpiada y consolidada en un solo patrón que será usado en el árbol de búsqueda. El resultado de la búsqueda devuelve una serie de identificadores de las jugadas que mayor posibilidad tienen de ser ejecutadas en el momento. La medida de similitud empleada para determinar las mejores posibilidades se basa primero en la diferencia de distancias entre la situación actual y los patrones en la base de conocimiento, como segundo criterio se usa el número de jugadores que participan en la jugada, dando así, mayor relevancia a los patrones que involucren una mayor cantidad de agentes.

El proceso de recuperación de patrones se encuentra basado en el Razonamiento Basado en Casos (RBC). RBC es una técnica de aprendizaje automático que se basa en el pensamiento humano y su forma para reaccionar a situaciones actuales basado en experiencias previas. Este tipo de técnica ha sido empleado previamente en este dominio [10,11]. Al conjuntar los patrones de juego que coinciden con la situación actual del campo de juego, se obtiene la ruta que el balón sigue en cada uno de ellos y se hace una predicción con un muestreo de 5 ciclos. Esto nos deja con una serie de probabilidades distribuidas a lo largo del tiempo sobre la posición del balón, al conjuntarse se puede obtener un patrón de predicción como muestra la Figura 3.

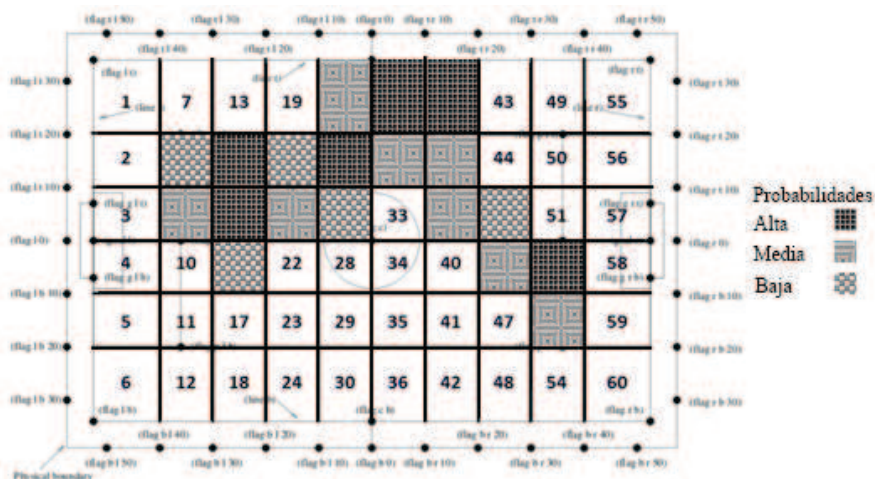


Figura 3. Probabilidad de la presencia del balón a lo largo de una predicción.

Teniendo una predicción con un muestreo de 5 ciclos nos permite determinar las zonas que deben ser cubiertas por nuestros jugadores para evitar el avance del rival y cuales tienen menor prioridad. Siendo un juego de fútbol, las zonas que rodean a la zona predicha también deben ser cubiertas para impedir completamente el ataque del contrario.

Algunas consideraciones tomadas en cuenta dentro del juego al llevar a cabo la predicción son:

- Se debe considerar si es viable mandar a los jugadores a defender una zona de cobertura dependiendo del tiempo pronosticado y la distancia de los jugadores a dicha zona.
- Se debe tener en cuenta que el proceso de predicción toma entre 1 y 2 ciclos para ser completado, sin embargo, en casos extremos en los que no se encuentre un patrón similar se puede tardar hasta 15 ciclos por lo que debe determinarse si se debe continuar la predicción o si es necesario establecer un límite de ciclos.
- En ocasiones en las que el balón se encuentre cercano a la portería, es recomendable deshabilitar la predicción para permitir que el portero se encargue de llevar a cabo sus labores naturales.

Resueltas las condiciones anteriores, el estado natural del juego permite al portero determinar la viabilidad de tomar acciones defensivas, las debe comunicar al resto del equipo y consisten en las zonas y el tiempo en el que un espacio debe ser cubierto puesto que se considera que el balón rondará el lugar.

Los registros generados en las pruebas pueden ser analizados e incluidos en la base de conocimiento. Esto permite al sistema evolucionar y aprender situaciones en las que los rivales han empezado a adaptarse a las condiciones generadas por la predicción.

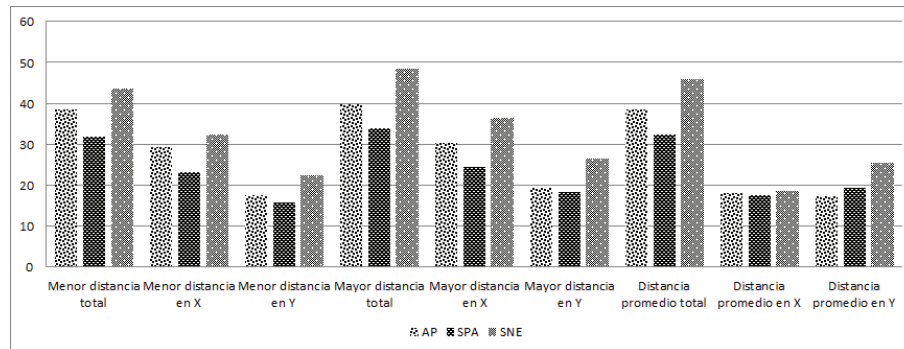


Figura 4. Distancias para: AP – Analizados Previamente; SPA – Sin Previo Análisis; SNE – Sin Nuestro Equipo.

4. Resultados

Se realizaron tres tipos de pruebas: nuestro equipo contra equipos previamente analizados e incluidos en la base de conocimiento, nuestro equipo contra equipos no incluidos en la base de conocimiento y juegos en los que nuestro equipo no participó. El primer tipo de pruebas se usó para probar la confiabilidad del SPPJ frente a equipos conocidos. La segunda muestra la confiabilidad sobre situaciones desconocidas para el sistema. El tercer tipo es una prueba general para mostrar la utilidad de la aplicación fuera de las estrategias empleadas por un solo equipo. Se ejecutaron 27 pruebas de las cuales 12 fueron del primer tipo, 3 sobre el segundo y 12 más sobre el tercero.

En general, el SPPJ logró obtener una precisión en la predicción de la posición del balón de más del 80 % en un rango aceptable, siendo éste de una zona que rodea la posición real del balón. Esto permite definir las zonas de cobertura en un tiempo en el que puedan ser ocupadas por nuestros jugadores para que el ataque sea detenido y el balón recuperado.

Los resultados que involucran la distancia entre la predicción y la realidad se muestran en la Figura 4. Según lo obtenido, tomar las distancias promedio en X y en Y brindan las mejores posiciones para determinar la situación actual.

La Figura 5 muestra el porcentaje de resultados que son útiles para la predicción empleando sólo la distancia promedio en X y la distancia promedio en Y. En la mayoría de los casos excede al 80 % por lo que se puede tener cierta fiabilidad a la hora de llevar a cabo las acciones defensivas.

Analizando los resultados mostrados, pudimos observar las siguientes situaciones:

- Existen equipos que su comportamiento no es consistente con lo obtenido en la base de conocimiento por lo que la predicción de su movimiento se dificulta.
- En las predicciones, existen situaciones en las que el balón no llega a las zonas en las que se suponía debía pasar debido a situaciones que se presentaron

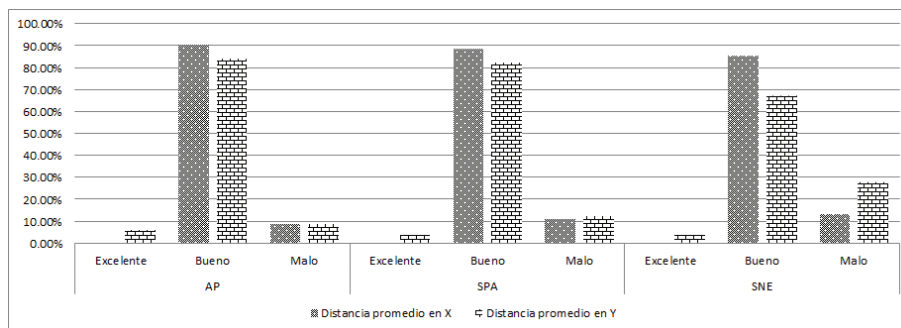


Figura 5. Porcentaje de utilidad en los resultados para: AP – Analizados Previamente; SPA – Sin Previo Análisis; SNE – Sin Nuestro Equipo.

en la ruta. Por ejemplo en algunos casos se cambia la ruta al presentarse un jugador de nuestro equipo como obstáculo y otras en las que nuestro equipo logra recuperar el balón.

- Algunos equipos cambian sus estrategias y tácticas dependiendo del resultado del juego y del tiempo que queda por jugar.

Aun teniendo en cuenta estas situaciones, se considera que las predicciones tienen un buen rango de precisión y se logran llevar a cabo en un tiempo suficiente para reaccionar.

5. Conclusiones y trabajo a futuro

En este artículo, se presentó el SPPJ, un sistema que permite obtener información de los registros de partidos del Simulador 2D de RoboCup y conjunta en una base de conocimientos los patrones de estrategias usadas por oponentes. Se genera un árbol de búsqueda que indexa todos los posibles casos para poder ser comparados en tiempo real. En tiempo de juego, se usa la información correspondiente a la posición de los jugadores y el balón, se compara con los registros guardados y se genera una predicción de la posible ruta que el balón seguirá. Todo el proceso se lleva a cabo en un tiempo muy corto para que pueda emplearse en tiempo real aun en el ambiente dinámico en el que fue probado.

El modelo discutido obtiene una precisión de resultados útiles de más del 80 % en las pruebas. Dadas las consideraciones sobre el ambiente, podemos concluir que es un porcentaje muy aceptable.

Se considera que el modelo presentado puede ser llevado a uso en otro tipo de dominios que no requieran una competición directa, dominios en los que se requiera obtener un patrón claro y un objetivo a predecir. También se requiere tener un registro de experiencias previas para poder llevar a cabo la comparación.

Las mejoras y el trabajo a futuro incluyen:

- Encontrar la división óptima de las zonas.

- Incorporación de pesos en la fase de aprendizaje.
- Emplear distintas técnicas de aprendizaje supervisado.

Referencias

1. Wooldridge, M.: An introduction to multiagent systems. Wiley (2008)
2. Del Giudice, A., Gmytrasiewicz, P., Bryan, J.: Towards strategic kriegspiel play with opponent modeling. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2. AAMAS '09, Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems (2009) 1265–1266
3. Laviers, K., Sukthankar, G., Molineaux, M., Aha, D.W.: Improving offensive performance through opponent modeling. In: Proceedings of Artificial Intelligence for Interactive Digital Entertainment Conference (AIIDE). (2009) 58–63
4. Richards, M., Amir, E.: Opponent modeling in scrabble. In: Proceedings of the 20th international joint conference on Artificial intelligence. IJCAI'07, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2007) 1482–1487
5. Kuhlmann, G., Knox, W.B., Stone, P.: Know thine enemy: a champion robocup coach agent. In: proceedings of the 21st national conference on Artificial intelligence - Volume 2. AAAI'06, AAAI Press (2006) 1463–1468
6. Parker, A., Nau, D., Subrahmanian, V.S.: Overconfidence or paranoia? search in imperfect-information games. In: proceedings of the 21st national conference on Artificial intelligence - Volume 2. AAAI'06, AAAI Press (2006) 1045–1050
7. Stone, P., Riley, P., Veloso, M.: Defining and using ideal teammate and opponent models. In: Proceedings of the Twelfth Annual Conference on Innovative Applications of Artificial Intelligence. (2000)
8. McCracken, P., Bowling, M.: Safe strategies for agent modelling in games. In: AAAI Fall Symposium on Artificial Multi-agent Learning. (2004) 103–110
9. RoboCup: Robocup website (2009)
10. Arias, M.: Definición de una arquitectura para la generación y selección de estrategias en equipos de fútbol robótico. Master's thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Estado de México (2008)
11. Berger, R., Burkhard, H.D.: At humboldt team description 2007. Lecture Notes in Computer Science (RoboCup 2007) (2008)

Análisis del desempeño de los algoritmos Redes Neuronales de Impulso (SNN) y Máquinas de Soporte Vectorial (SVM) en programación paralela

Israel Tabarez Paz, Neil Hernández Gress y Miguel González Mendoza

Universidad Autónoma del Estado de México
Atizapán de Zaragoza, México
{itabarezp}@uaemex.mx
<http://www.uaem.mx/cuyuaps/vallemexico.html>
Tecnológico de Monterrey, Campus Estado de México,
Atizapán de Zaragoza, México
{ngress,mgonza}@itesm.mx
<http://www.itesm.edu>

Resumen En este artículo se analizará el desempeño y la velocidad de entrenamiento de los algoritmos Redes Neuronales de Impulso (SNN) y Máquinas de Soporte Vectorial (SVM). La programación de SNN y SVM se realizan en arquitectura GPU NVIDIA modelo GeForce 9400M. Esto es aplicado a la clasificación de bases de datos. Acerca de la literatura, ningún autor ha codificado SVM QP (Programación Cuadrática) en un GPU, sin embargo existen otros métodos de optimización de SVM utilizando un GPU, tales como SVM light y SMO. En el caso de SNN, éste modelo se ha desarrollado en MatLab, FPGAs o circuitos secuenciales, pero ningún autor lo ha codificado en un GPU. Los resultados nos muestran que no en todos los casos es mejor en un GPU que en CPU, lo cual depende de la complejidad algorítmica y la cantidad de recursos computacionales requeridos. Desafortunadamente, en la actualidad la mayoría del hardware específico para programación paralela tiene memoria muy limitada para grandes bases de datos. Por esta razón, el programador tiene que reducir el grado de paralelización.

Palabras clave: GPU, FPGA, redes neuronales artificiales, redes neuronales de impulso, máquinas de soporte vectorial.

1. Introducción

Este artículo presenta una investigación acerca del análisis del desempeño de las SVMs en comparando con las SNN para la solución de problemas como clasificación (Herrero Lopez [8], reconocimiento de patrones [29], regresión (Carpenter [19], y construcción de Redes Neuronales Artificiales en un hardware específico, como las FPGAs (Papadonikolakis [7]. Respecto a la programación

paralela [39], existen diferentes formas: nivel bit (microcontrolador), nivel instrucción o dato (GPU), y nivel tarea (como MPI). En la figura 1 podemos ver el principio del cómputo paralelo. Los diversos problemas pueden ser resueltos en lenguajes CUDA, OpenGL, Cg, C, C++ and FORTRAN.

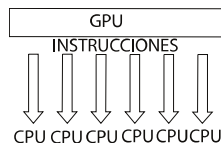


Figura 1. Ejecución de instrucciones en paralelo.

Este artículo está distribuido en las siguientes secciones: la sección 2 contiene estado del arte, la sección 3 consiste en la fase experimentación para comparación de estas metodologías y conclusiones en la sección 4.

2. Estado del arte

Sridhar [1], propone la simulación térmica de circuitos integrados a través de redes neuronales artificiales (RNA) sobre una GPU. El tiempo de aprendizaje es comparado entre una CPU y una GPU para esta aplicación. Asimismo, este autor señala que la principal ventaja de los GPU sobre los CPU es el alto paralelismo y eficiencia con un bajo costo. Sin embargo, nosotros probaremos que sobre las redes neuronales SVM y SNN, el costo computacional es alto y la eficiencia se reduce debido a las pérdida de precisión. También, que aunque existen circuitos integrados para alto paralelismo, es muy difícil trasladar este paralelismo a un software más eficiente.

Prabhu [21], compara la eficiencia del cerebro humano con el enorme poder computacional de los GPU's, aunque éste tiene limitaciones. El autor comenta que la principal función de los GPU's está en aprovechar las RNA. También los GPU's han sido usados para renderizar imágenes de alta calidad en tiempo real, realidad virtual y video juegos. Los GPU's modernos pueden ejecutar tareas en paralelo en grandes cantidades.

De acuerdo con Sergio Herrero et al. [8] el tiempo de entrenamiento de una tarea binaria compuesta de 100,000 puntos con cientos de dimensiones puede tardar horas para la ejecución serial. Sin embargo, es decir, el GPU es más lento para alguna ejecución serial que el CPU; y el GPU es más rápido para ejecuciones en paralelo que los CPU's.

Finalmente, Benjamin Schrauwen [25] propone a contribuye a mejorar el algoritmo spikeprop ajustando los parámetros umbral, retardo y derivada del tiempo. Esto puede ayudarnos a seleccionar las constantes óptimas para nuestros experimentos.

2.1. Programación paralela de las RNA

La taxonomía de paralelización se aproxima a la neurosimulaciones representadas en la figura 2, esto es de acuerdo con Nikola [35].

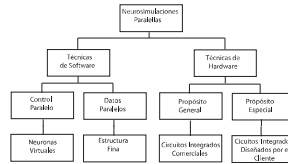


Figura 2. Taxonomía de paralelización para neurosimulaciones.

Las RNA (Redes Neuronales Artificiales) pueden ser paralelizadas en hardware específico como GPU's, FPGA como Thomas [18], circuitos secuenciales, o tarjetas electrónicas específicas [14]. Sin embargo, nos hemos enfocado al modelo GeForce 9400M. En caso de los SVM's, pueden ser algoritmos diseñados tanto en GPU's o CPU's debido al método de optimización requerido para resolver operaciones de matrices repetitivamente. Sin embargo, en el caso de SNN, es una metodología que está orientada más para resolverse en FPGA o circuitos secuenciales, aunque puede ser simulado en un GPU, esto porque es uestá enfocada más a tiempo real; por otro lado su función de activación se puede resolver con aproximaciones matemáticas, de tal manera que si el número de neuronas en la capa de entrada aumenta, entonces también aumenta la complejidad computacional. A pesar de esto, si aplicamos circuitos secuenciales, el umbral puede ser detectado con un detector de nivel de voltaje, que es un circuito muy simple de resolver.

Redes neuronales de impulso (SNN) La arquitectura de las SNN consiste en un conjunto de nodos o neuronas que contienen múltiples retardos como se observa en la figura 3 (Olaf [29]).

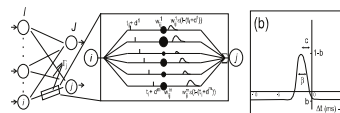


Figura 3. Conectividad de las SNN.

Formalmente, una neurona j , que pertenece a un conjunto Γ_j ('neurona pre-synaptica') Olaf [29], recibe un conjunto de impulsos que son disparados en el tiempo, es decir, $t_i, i \in \Gamma_j$ cuando la terminal alcanza un umbral θ . La dinámica de la variable interna $x_j(t)$ está determinada por la función de respuesta del impulso $\varepsilon(t)$ donde w_{ij} son los pesos de la conexión de la i a la j :

$$x_j(t) = \sum_{i \in \Gamma_j} w_{ij} \varepsilon(t - t_i) \quad (1)$$

El potencial sináptico está definido por la ecuación (2):

$$x_j(t) = \varepsilon(t) = \frac{t}{\tau} e^{1 - \frac{t}{\tau}} \quad (2)$$

El retardo d_k de la terminal k está definido por la diferencia entre el tiempo de disparo de la neurona presináptica y el tiempo de subida de la neurona post sináptica. La constante τ es una constante de tiempo que controla el tiempo de subida y el tiempo de caída. Así, múltiples sinapsis por conexión de la neurona que recibe j están descritas por la siguiente ecuación (3):

$$x_j(t) = \sum_{i \in \Gamma_j} \sum_{k=1}^m w_{ij}^k y^k(t) \quad (3)$$

Para una terminal de salida, se describe la siguiente ecuación:

$$y_j^k(t) = \varepsilon(t - t_i - d^k) \quad (4)$$

El Modelo de Respuesta Simple asegura la causalidad, por lo que $\eta(s) = 0$ para $s \leq 0$ y usualmente no positivo, se describe por la ecuación (5):

$$\eta(s) = -\vartheta e^{\frac{s}{\tau_r}} H(s) \quad (5)$$

Donde θ es el umbral de la neurona, $H(s)$ es la función escalón unitario τ_r es otra constante de tiempo.

Por lo tanto el potencial de una neurona $u_j(t)$ es:

$$u_j(t) = \sum_{t_j^{(f)} \in F_j} \eta(t - t_j^{(f)}) + \sum_{i \in \Gamma_j} \sum_{t_i^{(g)}} w_{ij} \varepsilon(t - t_i^{(g)} - d_{ij}) \quad (6)$$

Donde

$$F_j = \{t^{(f)}; 1 \leq f \leq n\} = \{t | u_j(t) = \vartheta\} \quad (7)$$

Y , n denota el número de impulsos.

Donde t tiene que ser definido en el GPU como un intervalo de tiempo. En este caso los bloques contienen 255 hilos, pero en otros modelos de GPU's NVIDIA este valor es de 512 o 1024, lo que depende de las especificaciones técnicas. En nuestro caso disponemos de 60000 hilos distribuidos en 235 bloques con 255 hilos cada uno, aproximadamente. El paralelismo en SNN es implementado en bloques configurados en 3D como sigue:

$$threadx = \frac{SIZE}{NN[num_cap] * NN[num_cap + 1]} \quad (8)$$

Donde $threadx$ es el tiempo t ; $SIZE$ es el número de hilos por bloque; $NN[num_cap]$ es el número de neuronas en la capa i ; $NN[num_cap + 1]$ es el número de neuronas en la siguiente capa j . Por otro lado, podemos observar que los retardos no se toman en cuenta al paralelizar los bloques, ya que éstos son aplicados de manera secuencial debido a las limitaciones de memoria; de la tarjeta NVIDIA. Asimismo, si el tiempo excede el número de hilos por bloque entonces necesitamos agregar más bloques, sin embargo las áreas entre bloques son codificadas secuencialmente. Esto consume mucho tiempo de ejecución, el cual afecta al tiempo de aprendizaje de la red neuronal. Esta configuración se puede representar por un cubo como el de la figura 4. Los ejes horizontales representan el peso de los retardos y los valores de la exponencial de tiempo, respectivamente; y el eje vertical representa el número de neuronas de la capa anterior.

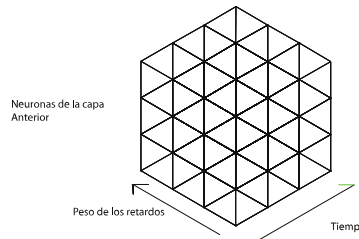


Figura 4. Paralelización de la SNN (3D).

Máquinas de soporte vectorial (SVM) La paralelización de fase de aprendizaje en el caso de SVM QP consiste en la simplificación del tiempo de ejecución de las operaciones matriciales y vectoriales que implica el mismo algoritmo. Estas operaciones ya están resueltas en la página de NVIDIA, en la literatura misma [39], aunque esto depende del método de optimización aplicado, que en nuestro caso es QP (Programación cuadrática) [40]. Debido a la configuración de los bloques e hilos del GPU, esto nos da capacidad computacional para resolver matrices cuadráticas de 16, 22 o 32 datos de entrada respectivamente. Como resultado de esto, el GPU fue configurado en la mayoría de los casos con más de un bloque para cubrir todos los elementos de la base de datos. Lo que implicó la reducción de la paralelización a nivel de operaciones vectorial. Además, de considerar todos los bloques posibles para las bases de datos con máximo 1400 instancias aproximadamente, y la paralelización óptima para 16 datos, ya que es el tamaño máximo en hilos de un bloque. Por otro lado, a menor paralelismo del código implicó tener más de un kernel combinando código serial con

paralelo y copiando del GPU al CPU varias veces y viceversa, por la naturaleza del código. Esto afectó de manera importante a la metodología SVM. En este caso para más de 1400 instancias se consideró a la metodología SVM light. El modelo matemático de SVM QP está descrito como se muestra a continuación:

$$\begin{aligned}
 & \text{máximo } \alpha \\
 q(\alpha) &= 0,5\alpha^T Q\alpha - 1^T \alpha \tag{9} \\
 & \text{sujeto a} \\
 & \mathbf{y}^T \alpha = 0 \\
 & 0 \leq \alpha \leq C
 \end{aligned}$$

La matriz Q está compuesta por $(Q)_{ij} = y_i y_j k(x_i, x_j)$, donde:
 $i, j = 1, 2, 3, \dots, l$, $\alpha = [a^1 \dots a^l]^T$
 $\mathbf{1} = [1^1 \dots 1^l]^T$ $\mathbf{y} = [y^1 \dots y^l]^T$
 $C = [C^1 \dots C^l]^T$

Finalmente, la salida es $y = \text{sign}(\sum_{i=1}^N y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i))$. figura 5 representa la arquitectura de SVM.

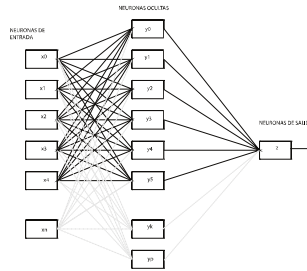


Figura 5. Arquitectura de SVM.

En este caso, fue paralelizada la multiplicación matricial para $((Q)_{ij} = y_i y_j k(x_i, x_j))$ en el método QP. El número de hilos fue generalizado como sigue:

$$\text{BlocksPerRow} = \frac{\text{NUMBER OF DATAS}}{\text{THREADS PER BLOCK}} + 1 \tag{10}$$

$$\text{ThreadPerRow} = \frac{\text{NUMBER OF DATAS}}{\text{BlocksPerRow}}$$

A pesar de eso la tarjeta NVIDIA tiene numero de hilos limitado, por lo tanto esa es la longitud máxima de un vector o número de datos copiados del CPU al GPU divididos entre el número de atributos por dato.

3. Experimentos

3.1. Redes neuronales de impulso (SNN)

En esta sección mostramos el resultado obtenido de la paralelización de SNN que se muestra en el cuadro 1.

Tabla 1. Resultados de algunas bases de datos para SNN.

Datos	Instancias	Atributos	Clases	Iteraciones	Exitos	Tiempo de Aprendizaje [ms]
Iris	150	4	3	12	149	52331.625
Cars	1729	6	4	12	1294	1319680.625
Breast Cancer Wisconsin	699	9	2	19	673	253460.0937
Adult	32561	14	2	30	29247	24 hrs aprox.
Heart Disease Cleveland (HDC)	303	13	5	40	221	1069932.75

Sin embargo, observamos que SNN converge exponencialmente hacia la solución, como en la figura 6 (caso XOR).

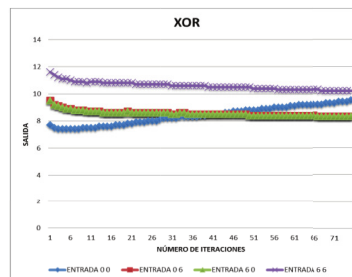


Figura 6. Solución del problema XOR.

En el cuadro 2 se muestra codificación y la solución numéricamente.

3.2. Máquinas de soporte vectorial (SVM)

La metodología SVM es usada para clasificación, regresión, predicción, y densidades de datos. En este caso lo aplicamos a clasificación de datos. Para una

Tabla 2. Caso XOR para SNN.

X1 ms	X2 ms	BIAS ms	SALIDA DESEADA ms
0	0	0	10
0	6	0	8
6	0	0	8
6	6	0	10

red SVM multiclase, aplicamos redes en serie, es decir que la salida de la primer red binaria es la entrada de la siguiente, considerando que se van clasificando por clases binarias. El número de redes en cascada se calcularon:

$$(RedSerial\ SVM) = (Clases) - 1 \quad (11)$$

Los resultados para QP - SVM se muestran en el cuadro 3.

Tabla 3. Resultados de algunas bases de datos para SVM.

Datos	Instancias	Atributos	Clases	Iteraciones	Exitos	Tiempo de Aprendizaje [ms]
Iris	150	4	3	2758	145	1521128.5
Cars (SVM light)	1728	6	4	—	1728 (Clase 1) 56 (Clase 2) 4 (Clase 3) 34 (Clase 4)	608927.125 1478488.5 1090742.125 970055.375
Breast Cancer Wisconsin	699	9	2	—	641 (Clase 1) 127 (Clase 2)	163790.55625 162216.453125
Adult	32561	14	2	—	—	—
Heart Disease Cleveland (HDC)	303	13	5	—	147 (Clase 1) 3 (Clase 2) 66 (Clase 3) 25 (Clase 4) 2 (Clase 5)	56749.73046 21624.41406 22715.71875 27424.23046 98908.94531

Los resultados de los experimentos, para bases de datos multiclase, el kernel de cada clase tiene que se encontrado ya que el hiperplano por cada clase a considerar cambia de dimensión al clasificar cada clase de la misma base de datos. Por ejemplo, en el caso de la base de datos Cars: la clase 1 de Cars utiliza el kernel “erbf 3”, la clase 2 con polinomial de grado 3, la clase 3 con polinomial de grado 6 y la clase 4 con polinomial de grado 8.

De esta manera se llega a los siguientes resultados del cuadro 4 al comparar los tiempos de aprendizaje de ambas metodologías.

Tabla 4. Comparación del tiempo de aprendizaje de SNN vs SVM.

Base de datos	Tiempo de Aprendizaje SNN [ms]	Tiempo de Aprendizaje SVM [ms]
Iris	152331.625	1521128.5
Cars	1319680.625	4148213.125
Breast Cancer Wisconsin (BCW)	253460.09375	326007.0156
Adult	86400000 aprox.	—
Heart Disease Cleveland (HDC)	1069932.75	227423.0906

En la figura 7 se grafica el cuadro 4 y podemos observar que en una arquitectura GPU SNN es más rápido que SVM. En el caso de la base de datos Adult, no es comparable ya que SVM no es la metodología adecuada en un GPU por el enorme tiempo de aprendizaje que consume. Para bases de datos más pequeñas con menos de 16 instancias, SVM es más rápido que SNN por el uso de un solo bloque con 256 hilos, con lo cual se paraleliza al máximo y dicha metodología converge más rápido que SNN. Debido a que el GPU no provee más de 60000 flops o 245 datos (aprox.) en contraste con la base de datos Adult, ya que esta requiere 1,060,218,721 flops para datos flotantes. Esto nos obliga a reducir la paralelización lo que implica más tiempo de procesamiento que un CPU. La causa es que SVM bajo esas condiciones requiere tomar en cuenta más de un kernel y copiar del CPU al GPU varias veces y viceversa, lo que consume enormes cantidades de tiempo.

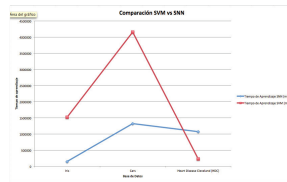


Figura 7. Comparación del Tiempo de Aprendizaje de SNN vs SVM.

Sin embargo, SVM es mayor para bases de datos binaria que SNN, aunque los resultados del cuadro 5 muestran que en un GPU SNN es más preciso que SVM.

En la figura 8 se muestra la gráfica de la comparación de la tabla anterior.

Tabla 5. Comparación de precisión SNN vs SVM.

Base de Datos	Exitos de SVM	Exitos de SNN
Iris	145	149
Cars (SVM light)	1728 (Clase 1) 56 (Clase 2) 4 (Clase 3) 34 (Clase 4)	1294 1294 1294 1294
Breast Cancer Wisconsin	641 (Clase 1) 127 (Clase 2)	673 673
Adult	—	29247
Heart Disease Cleveland (HDC)	147 (Clase 1) 3 (Clase 2) 66 (Clase 3) 25 (Clase 4) 2 (Clase 5)	221 221 221 221 221

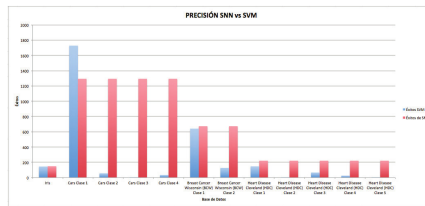


Figura 8. Comparación de precisión SNN vs SVM.

4. Conclusiones y trabajo futuro

Los resultados obtenidos muestran que SNN es más eficiente y veloz que SVM en una arquitectura GPU debido a que éste último requiere enormes cantidades de recursos computacionales. Las SVM convergen más rápido que las SNN pero en un GPU pierde precisión y por lo tanto SVM converge más lento en un GPU que en un CPU. Sin embargo, SNN no consume tantos recursos de memoria como SVM y por lo tanto es más rápida su ejecución a pesar de converger más lento que SVM. También SNN no requiere tanta precisión en la salida porque utiliza ventanas de tiempo para codificar la salida, por ejemplo si la ventana de tiempo va desde 10 a 12 un 10.0 o 11.5 puede representar un 1 lógico.

En el caso de SNN al aplicar algún circuito lógico y secuencial con un circuito de reloj, entonces su precisión puede ser afectada cuando la frecuencia de reloj sea alta. Sin embargo para aplicaciones reales la velocidad de los sensores podrían limitar la detección del umbral.

Como trabajo futuro es necesario implementar SNN en tiempo real con el hardware adecuado para otras aplicaciones. Por otro lado, SNN puede ser extendido para más capas con la finalidad de conocer los efectos sobre el resultado.

Referencias

1. Sridhar, A.; Vincenzi, A.; Ruggiero, M.; Atienza, D.; “Neural Network-Based Thermal Simulation of Integrated Circuits on GPUs, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on , vol.31, no.1, pp. 23–36, Jan. 2012.
2. Ahmadi, Arash; Soleimani, Hamid; ‘A GPU based simulation of multilayer spiking neural networks, Electrical Engineering (ICEE), 2011 19th Iranian Conference on , vol., no., pp. 1–5 17–19 May 2011.
3. Lowe, E.W.; Woetzel, N.; Meiler, J. “Poster: GPU-accelerated artificial neural network for QSAR modeling, Computational Advances in Bio and Medical Sciences (ICABS), 2011 IEEE 1st International Conference on , vol., no., pp. 254, 3–5 Feb. 2011.
4. B. Kirk, David; W. Hwu, Wen-mai “Programming Massively Parallel Processors. Ed. Morgan Kaufmann, 2010.
5. Qi Li; Salman, R.; Kecman, V.; “An intelligent system for accelerating parallel SVM classification problems on large datasets using GPU, Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on , vol., no., pp. 1131–1135, Nov. 29 2010–Dec. 1 2010.
6. [5] Tsung-Kai Lin; Shao-Yi Chien; “Support Vector Machines on GPU with Sparse Matrix Format,” Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on , vol., no., pp. 313–318, 12–14 Dec. 2010
7. Papadonikolakis, M.; Bouganis, C.: ‘A novel FPGA-based SVM classifier,’ Field-Programmable Technology (FPT), 2010 International Conference on , vol., no., pp. 283–286, 8–10 Dec. 2010
8. Sergio Herrero-Lopez, John R. Williams, Abel Sanchez,; ‘Parallel Multiclass Classification using SVMs on GPUs’, November 2010.
9. Izhikevich, E.M.: ‘Hybrid spiking models’, vol. 368, issue 1930, pp. 5061–5070, Nov 2010.
10. Venkittaraman Vivek, Pallipuram Krishnamani: ‘ACCELERATION OF SPIKING NEURAL NETWORKS ON SINGLE-GPU AND MULTI-GPU SYSTEMS’, May 2010.
11. Bhuiyan, M.A.; Pallipuram, V.K.; Smith, M.C.: “Acceleration of spiking neural networks in emerging multi-core and GPU architectures,” Parallel and Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on , vol., no., pp. 1–8, 19–23 April 2010.
12. Scanzio, S.; Cumani, S.; Gemello, R.; Mana, F.; Laface, P.; ‘Parallel implementation of artificial neural network training,’ Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on , vol., no., pp. 4902–4905, 14–19 March 2010.
13. Yudanov, D.; Shaaban, M.; Melton, R.; Reznik, L.; GPU-Based Simulation of Spiking Neural Networks with Real-Time Performance and High Accuracy, Feb 2010.
14. XIN JIN: Parallel Simulation of Neural Networks on Spinnaker Universal Neuro-morphic Hardware, University of Manchester, 2010.

15. Papadonikolakis, M.; Bouganis, C.-S.; Constantinides, G.: ‘Performance comparison of GPU and FPGA architectures for the SVM training problem,’ *Field-Programmable Technology*, 2009. FPT 2009. International Conference on , vol., no., pp. 388–391, 9–11 Dec. 2009.
16. Fidjeland, A.K.; Roesch, E.B.; Shanahan, M.P.; Luk, W.; ‘NeMo: A Platform for Neural Modelling of Spiking Neurons Using GPUs,’ *Application-specific Systems, Architectures and Processors*, 2009. ASAP 2009. 20th IEEE International Conference on , vol., no., pp. 137–144, 7–9 July 2009.
17. Nageswaran, J.M., Dutt, N.; Krichmar, J.L.; Nicolau, A.; Veidenbaum, A.; ‘Efficient simulation of large-scale Spiking Neural Networks using CUDA graphics processors,’ *Neural Networks*, 2009. IJCNN 2009. International Joint Conference on , vol., no., pp. 2145–2152, 14–19 June 2009.
18. Thomas, D.B.; Luk, W.: ‘FPGA Accelerated Simulation of Biologically Plausible Spiking Neural Networks,’ *Field Programmable Custom Computing Machines*, 2009. FCCM '09. 17th IEEE Symposium on , vol., no., pp. 45–52, 5–7 April 2009.
19. Carpenter, A.; ‘CUSVM: A CUDA IMPLEMENTATION OF SUPPORT VECTOR CLASSIFICATION AND REGRESSION’, Jan. 2009.
20. Stewart R.D., Bair W.: ‘Spiking neural network simulation: numerical integration with the Parker–Sochacki method’, Jan. 2009.
21. Prabhu, R.D.; ‘SOMGPU: An unsupervised pattern classifier on Graphical Processing Unit,’ *Evolutionary Computation*, 2008. CEC 2008. IEEE World Congress on Computational Intelligence. IEEE Congress on , vol., no., pp. 1011–1018, 1–6 June 2008.
22. Bryan Catanzaro, Narayanan Sundaram, Kurt Keutzer: *Fast Support Vector Machine Training and Classification on Graphics Processors*, International Conference on Machine Learning, Helsinki, Finland, 2008.
23. Martínez Z. M., Díaz P. F. J., Díez H. J. F, Antón R. M.: *Fuzzy ART Neural Network Parallel Computing on the GPU*, Springer (2007).
24. Philipp S., Grbl A., Meier K., Schemmel J.: *Interconnecting VLSI Spiking Neural Networks Using Isochronous Connections*, Springer (2007).
25. Benjamin Schrauwen, Jan Van Campenhout, *Improving SpikeProp: Enhancements to An Error-Backpropagation Rule for Spiking Neural Networks*, Ghent University ELIS-PARIS (2006)
26. L. Zhongwen, L. Hongzhi and W. Xincui: *Artificial Neural Network Computation on Graphic Process Unit* (2005).
27. N.G. Pavlidis, D.K. Tasoulis, V.P. Plagianakos, G. Nikiforidis, M.N. Vrahatis: *Spiking Neural Network Training Using Evolutionary Algorithms*, IEEE International Joint Conference on , pp. 2190 - 2194, vol. 4 ,december 2005.
28. Seijas Fossi C., Caralli D’ Ambrosio A.: *Uso de las máquinas de soporte para la estimación del potencial de acción celular*, *Revista de Ingeniería UC*. Vol. 11, N 1, 56- 61 (2004).
29. Olaf Booij,: *Temporal Pattern Classification using Spiking Neural Networks*, *Intelligent Sensory Information Systems Informatics Institute Faculty of Science Universiteit van Amsterdam* (2004).
30. Sander M. B.: *Spiking Neural Networks*, *Universiteit Leiden* (2003).
31. Izhikevich, E.M.: ‘Simple model of spiking neurons,’ *Neural Networks*, *IEEE Transactions on* , vol.14, no.6, pp. 1569– 1572, Nov. 2003
32. Sander M. Bohte, Joost N. Kok, Han La Poutre: *Error-backpropagation in temporally encoded networks of spiking neurons*, *Neurocomputing* 48, pp. 17 - 37, (2002).

33. Platt C. J.: Sequential Minimal Optimization: A fast Algorithm for Training Support Vector Machines (1998).
34. Osuna E., Freund R., Girosi F.: An Improved Training Algorithm for Support Vector Machines, In Proc. of IEEE NNSP'97.
35. Nikola B. Serbedijja: Simulating Artificial Neural Networks on Parallel Architectures (1996).
36. Vapnik V., Cortes C., Support Vector Networks (1995).
37. Fauset Laurene, Fundamentals of Neural Networks, ARCHITECTURE, ALGORITHMS, AND APPLICATIONS, Prentice Halls, 1994.
38. A. L. Hodgkin and A. F. Huxley: A quantitative description of membrane current and its application to conduction and excitation in nerve, J. Physiol. (1952).
39. <http://www.nvidia.com>
40. <http://www.svms.org>

Estudio comparativo de estrategias de control no lineal basadas en pasividad para sistemas robóticos

Miguel O. Uribe H., José M. Fernández R., Juan D. Ramírez Z.,
Luis R. Lechuga G., Omar A. Dominguez R. y Luis E. Ramos V.

Universidad Autónoma del Estado de Hidalgo, Centro de Investigación en
Tecnologías de Información y Sistemas, Mineral de la Reforma, Hidalgo, México
{m_oss_84, josmanuelfer, juanda_zam, l_1123581321,
lramosvelasco}@hotmail.com, omar@uaeh.edu.mx

Resumen El presente trabajo de investigación propone estrategias de control de movimiento libre, de naturaleza no lineal y basadas en pasividad, aplicadas a un sistema robótico para guiado háptico con propósitos de entrenamiento. Las estrategias de control de movimiento libre implementadas y comparadas de manera experimental en este trabajo, están basadas en la teoría de sistemas disipativos (pasividad), así como teoría de estabilidad para sistemas dinámicos de tipo no lineal. El estudio comparativo corresponde a la evaluación de leyes de control con compensación de gravedad y par calculado, en seguimiento de trayectorias.

Palabras clave: control no lineal, energía potencial, pasividad, guiado háptico, estabilidad, control par calculado, control PD Plus.

1. Introducción

Una interfaz háptica es un medio por el cual un operador humano interactúa con un ambiente virtual, mediante un dispositivo háptico, el cual en respuesta recibe señales de retroalimentación de fuerza[?].

Todo dispositivo háptico es clasificado de acuerdo a la forma de interacción humana y retroalimentación de fuerza. Algunos ejemplos de estos dispositivos son las plataformas de movimiento, guantes, dermatoesqueletos y de tipo mayordomo [?].

Phantom Premium 1.0, es un dispositivo háptico de tipo mayordomo. Posee 3 grados de libertad actuados y hasta seis sensorizados, articulaciones de revolución, cuyo extremo final es un dedal. Tiene como función primordial conceder a un operador humano una interacción con un ambiente virtual, percibiendo fuerzas de retroalimentación para el estímulo kinestésico[?].

La tarea de guiado háptico, al recibir consignas de operación como posición

y fuerza, obligan al dispositivo háptico a ser empleado como robot manipulador, con dinámica inercial y tribológica despreciable[?]. Debido a que los robots manipuladores en su totalidad cuentan con dinámicas no lineales, son considerados sistemas sofisticados de control que deben satisfacer requisitos tales como precisión y rapidez, motivo por el cual las leyes de control implementadas a estos dispositivos deben de ser de naturaleza no lineal, y compensar de manera eficiente la incertidumbre y las perturbaciones relacionadas con el funcionamiento del sistema[?].

Los controles clásicos corresponden a estructuras lineales con la dificultad de lograr regulación en cualquier configuración del espacio de trabajo de un robot.

La compensación de gravedad representa compensación de no linealidades alrededor del punto de equilibrio y permite la convergencia deseada, sin embargo, la energía que implica la compensación puede tener efectos contraproducentes en la vida útil de un robot, por lo que es conveniente realizar un estudio comparativo de diferentes controles y definir su desempeño bajo las mismas condiciones de operación, a partir de la comparación de error de posición articular, par de entrada global y energía total.

Por lo tanto el objetivo del experimento es evaluar de manera experimental dos técnicas de control articular para robots manipuladores, y definir el desempeño de un control compensado parcialmente con dinámica con relación a un control par calculado en tiempo real del robot.

El presente artículo tiene las siguientes contribuciones :

- Validación experimental de la dinámica gravitacional para la compensación durante la ejecución de la tarea, a través del control PD plus.
- Validación experimental de la dinámica inercial y de Coriolis a través de la aplicación del control par calculado pasivo robusto.
- Diseño y prueba de estabilidad de las leyes de control en competencia
- Prueba experimental de estabilidad y análisis de energía

En este artículo en la sección II se describe el control PD+G, en la sección III se hace la descripción general del control par calculado, la sección IV contiene la descripción de la tarea del estudio comparativo, así como los resultados obtenidos de cada control aplicado a la interfaz háptica, la parte V son las conclusiones del experimento, y finalmente los agradecimientos y las referencias.

2. Control PD+G

En esta sección se evalúa la estrategia de control PD plus sobre el dispositivo háptico PHANToM 1.0. A partir del modelo dinámico del sistema y de sus propiedades, se implementa la ley de control a comparar. El modelo dinámico

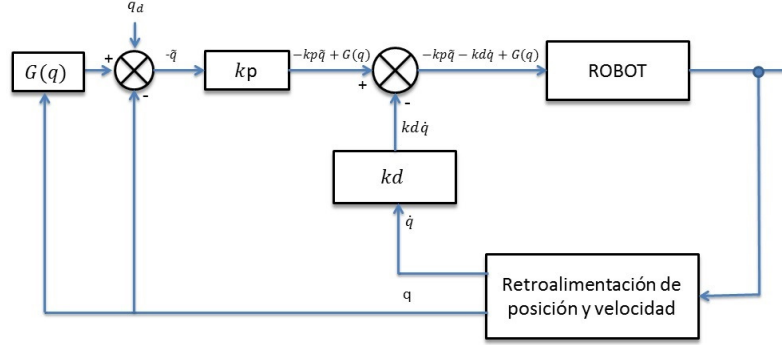


Figura 1. Diagrama de bloques del control PD+G.

reducido de este dispositivo es obtenido por medio del método de Euler-Lagrange y se define por

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (1)$$

donde $H(q) \in \mathbb{R}^{3 \times 3}$ es la matriz de inercias, $C(q, \dot{q}) \in \mathbb{R}^{3 \times 3}$ representa a la matriz de fuerzas centrípetas y de Coriolis, $G(q) \in \mathbb{R}^{3 \times 1}$ representa al vector de fuerzas debido a efectos gravitacionales, $\tau \in \mathbb{R}^{3 \times 1}$ vector de pares de entrada, q, \dot{q} y $\ddot{q} \in \mathbb{R}^{3 \times 1}$ vectores de posiciones, velocidades y aceleraciones articulares.

La matriz $H(q)$ es simétrica, y definida positiva, de tal manera que

$$H(q) = H(q)^\top, \\ \dot{q}^\top H(q) \dot{q} > 0; \forall q \in \mathbb{R}^{n \times 1}$$

La matriz $C(q, \dot{q})$ se relaciona con la matriz de inercias $H(q)$ de la siguiente manera

$$\dot{q}^\top \left\{ \dot{H}(q) - 2C(q, \dot{q}) \right\} \dot{q} \equiv 0; \forall x \in \mathbb{R}^{n \times 1},$$

que es la propiedad de antisimetría.

La ley de control PD+G está dada por

$$\tau = -K_p \tilde{q} - K_d \dot{\tilde{q}} + G(q), \quad (2)$$

donde $\tilde{q} = q - q_d \in \mathbb{R}^{3 \times 1}$, corresponde al error de posición articular y $\dot{\tilde{q}} = \dot{q} - \dot{q}_d \in \mathbb{R}^{3 \times 1}$, es el error de velocidad articular.

Para regulación se toman en cuenta las siguientes consideraciones

$$q_d = cte$$

$$\dot{q}_d = 0,$$

entonces sustituyendo los valores de q_d y \dot{q}_d , la ley de control queda definida de la siguiente manera

$$\tau = -k_p \tilde{q} - k_d \dot{\tilde{q}} + G(q), \quad (3)$$

donde $k_p \in \mathbb{R}^{3 \times 3}$ matriz de ganancias proporcionales, $k_d \in \mathbb{R}^{3 \times 3}$ matriz de ganancias derivativas. Ambas matrices definidas positivas.

Sustituyendo el valor de τ en (1) para el lazo cerrado se obtiene

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = -K_p \tilde{q} - K_d \dot{\tilde{q}} + G(q)$$

$$H(q)\ddot{q} + (C(q, \dot{q}) + K_d)\dot{q} + K_p \tilde{q} = 0 \quad (4)$$

en un equilibrio $\dot{q} = \ddot{q} = 0$, tal que $\tilde{q} = 0$ y $q = q_d$.

Para determinar la fuerza aplicada en los actuadores se necesita hacer una transformación de par de entrada a fuerza. Por lo tanto si el par esta dado por

$$\tau = J^T f,$$

entonces la fuerza queda definida de la siguiente manera

$$f = J^{-T} (-k_p \tilde{q} - k_d \dot{\tilde{q}} + G(q)), \quad (5)$$

donde J^{-T} es el Jacobiano inverso transpuesto y f la fuerza aplicada en el efector final.

3. Control par calculado

En esta sección se describe el diseño y prueba de estabilidad del control par calculado, implementado sobre el dispositivo háptico PHANToM 1.0. A partir del segundo método de Lyapunov se propone la siguiente función para obtener la ley de control

$$V = \frac{1}{2} \dot{\tilde{q}}^T H(q) \dot{\tilde{q}} + \frac{1}{2} \tilde{q}^T K_p \tilde{q}, \quad (6)$$

donde $\tilde{q} = q - q_d$, corresponde al error de posición articular y $\dot{\tilde{q}} = \dot{q} - \dot{q}_d$ es el error de velocidad articular.

El vector de fuerzas de Coriolis se representa por

$$\begin{aligned} C(q, \dot{q})\dot{\tilde{q}} &= C(q, \dot{q})\dot{q} - C(q, \dot{q})\dot{q}_d \\ C(q, \dot{q})\dot{q} &= C(q, \dot{q})\dot{\tilde{q}} + C(q, \dot{q})\dot{q}_d, \end{aligned} \quad (7)$$

con $C(q, \dot{q}) \in \mathbb{R}^{3 \times 3}$, $\dot{q} \in \mathbb{R}^{3 \times 1}$ vector de velocidad articular, $\dot{\tilde{q}} \in \mathbb{R}^{3 \times 1}$ vector de

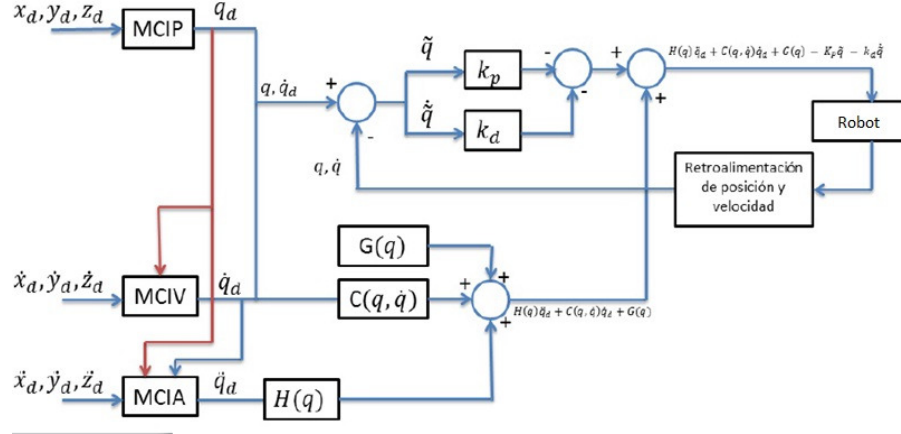


Figura 2. Diagrama de bloques del control par calculado.

error de velocidad articular, $\dot{q}_d \in \mathbb{R}^{3 \times 1}$ vector de velocidad articular deseada.

La matriz de fuerzas centrípetas y de Coriolis tiene la siguiente relación con la matriz de fuerzas de inercia

$$\dot{q}^T [\dot{H}(q) - 2C(q, \dot{q})] \dot{q} = 0, \quad \forall q, \dot{q} \in \mathbb{R}^n$$

y esta propiedad es mejor conocida como propiedad de anti-simetría.

Derivando (6) y sustituyendo el valor del vector de fuerzas centrípetas y de Coriolis descrito en (7) se obtiene

$$\begin{aligned} \dot{V} &= \dot{q}^T H(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{H}(q) \dot{q} + \dot{q}^T K_p \tilde{q}. \\ \dot{V} &= \dot{q}^T \tau - \dot{q}^T C(q, \dot{q}) \dot{q} - \dot{q}^T G(q) - \dot{q}^T H(q) \ddot{q}_d + \frac{1}{2} \dot{q}^T \dot{H}(q) \dot{q} - \frac{1}{2} \dot{q}^T \dot{H}(q) \dot{q}_d \\ &\quad + \dot{q}^T K_p \tilde{q}. \\ \dot{V} &= \dot{q}^T \tau - \dot{q}^T C(q, \dot{q}) \dot{q} - \dot{q}^T C(q, \dot{q}) \dot{q}_d - \dot{q}^T G(q) - \dot{q}^T H(q) \ddot{q}_d + \frac{1}{2} \dot{q}^T \dot{H}(q) \dot{q} \\ &\quad - \frac{1}{2} \dot{q}^T \dot{H}(q) \dot{q}_d + \dot{q}^T K_p \tilde{q}. \end{aligned} \quad (8)$$

Por la propiedad de antisimetría (8) queda definida por

$$\dot{V} = \dot{q}^T \tau - \dot{q}^T H(q) \ddot{q}_d - \dot{q}^T C(q, \dot{q}) \dot{q}_d - \dot{q}^T G(q) + \dot{q}^T K_p \tilde{q}. \quad (9)$$

Se usa la fricción viscosa con el propósito de disipar la energía almacenada en el dispositivo, y está representada de la forma

$$-\dot{\tilde{q}}^T K_d \dot{\tilde{q}}, \quad (10)$$

donde K_d es la matriz de ganancias derivativas definida positiva.

Finalmente usando (9) y (10) la ley de control obtenida está dada por

$$\tau = H(q)\ddot{q}_d + C(q, \dot{q})\dot{q}_d + G(q) - K_p \tilde{q} - K_d \dot{\tilde{q}}, \quad (11)$$

con $\tilde{q} = q - q_d \in \mathbb{R}^{3 \times 1}$ error de posición articular, $\dot{\tilde{q}} = \dot{q} - \dot{q}_d \in \mathbb{R}^{3 \times 1}$ error de velocidad articular, $K_p \in \mathbb{R}^{3 \times 3}$ matriz de ganancias proporcionales y $K_d \in \mathbb{R}^{3 \times 3}$ matriz de ganancias derivativas.

Para regulación se define lo siguiente

$$q_d = cte$$

$$\dot{q}_d = 0$$

$$\ddot{q}_d = 0$$

tomando en cuenta la consideraciones anteriores la ley de control queda de la forma

$$\tau = G(q) - K_p \tilde{q} - K_d \dot{\tilde{q}} \quad (12)$$

Podemos asumir, que el control par calculado pasivo robusto, es una generalización del control con retroalimentación dinámica y que garantiza seguimiento de trayectorias, y el control PD +G corresponde a un caso, útil en regulación global.

4. Experimentos

Se realizó la comparación del desempeño del control PD+G y el control par calculado, a partir del tiempo de convergencia a la trayectoria deseada, comparación del error de posición articular, el par de entrada global, medición de la energía total y la misma posición inicial en la coordenada (0,0,0).

La tarea consistió en regulación basada en seguimiento de una trayectoria determinada por las siguientes ecuaciones paramétricas

$$x_d = h + r \cos(\omega t)$$

$$y_d = 0,5$$

$$z_d = k + r \sin(\omega t)$$

donde $h = 0,02 \text{ m}$, $k = 0,03 \text{ m}$, $r = 0,02 \text{ m}$ y $\omega = 2\pi f$, que se designaron considerando el espacio de trabajo de Phantom Premium 1.0.

La velocidad y aceleración deseadas se definen por la primera y segunda derivada respectivamente de las ecuaciones paramétricas definidas anteriormente.

La frecuencia de muestreo $f = 1 \text{ KHz}$ y el tiempo de ejecución es de 10 segundos.

Se emplea el modelo cinemático directo e inverso de posición y velocidad para poder leer la posición operacional con los encoders del dispositivo háptico y aplicar un control articular a los actuadores.[?]

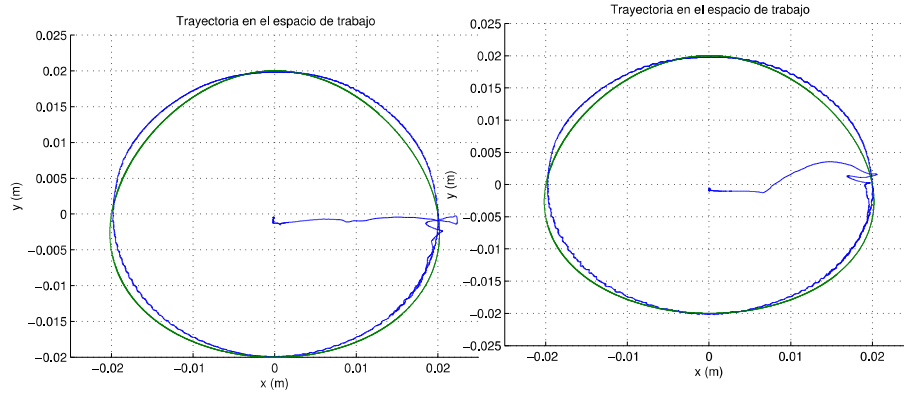


Figura 3. Trayectoria control PD+G.

Figura 4. Trayectoria control par calculado.

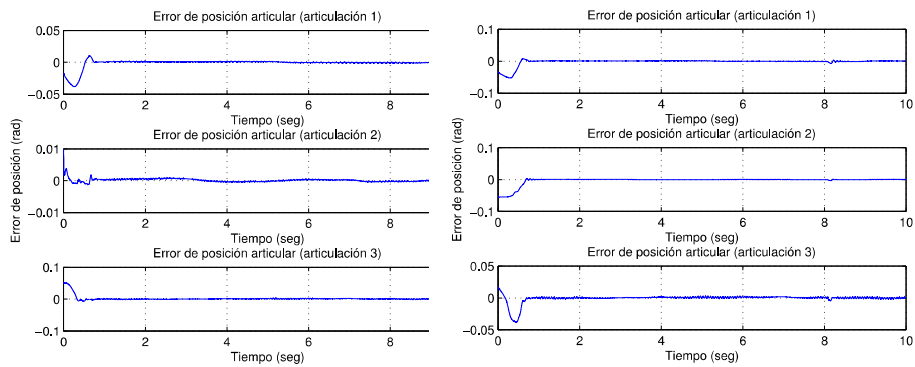


Figura 5. Error de posición con control PD+G.

Figura 6. Error de posición con control par calculado.

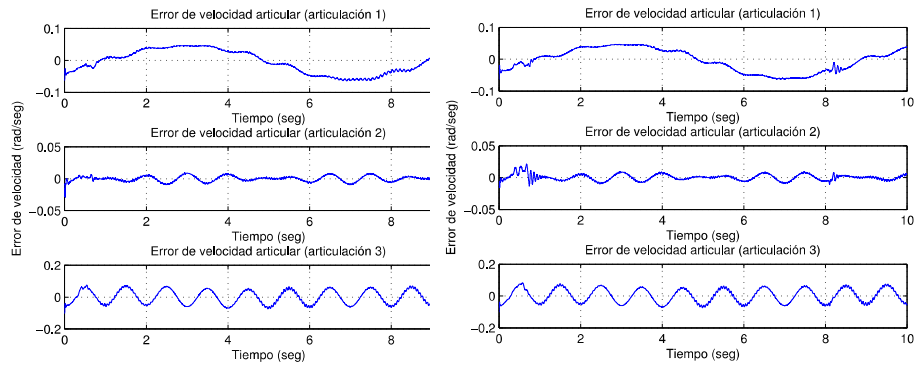


Figura 7. Error de velocidad con control PD+G.

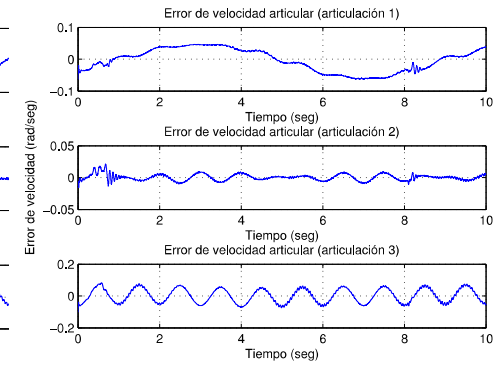


Figura 8. Error de velocidad con control par calculado.

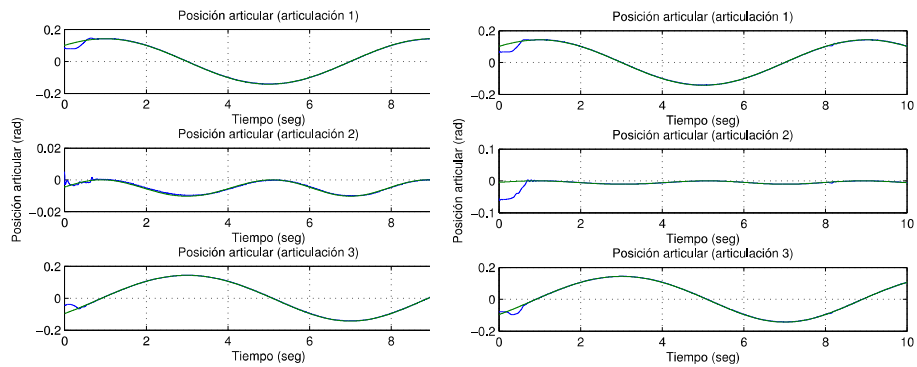


Figura 9. Posicion articular control PD+G.

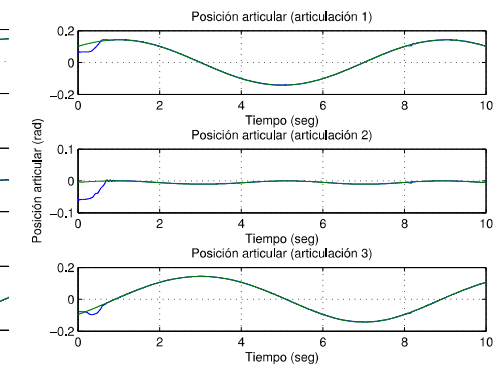


Figura 10. Posicion articular control par calculado.

Estudio comparativo de estrategias de control no lineal basadas en pasividad para sistemas robóticos

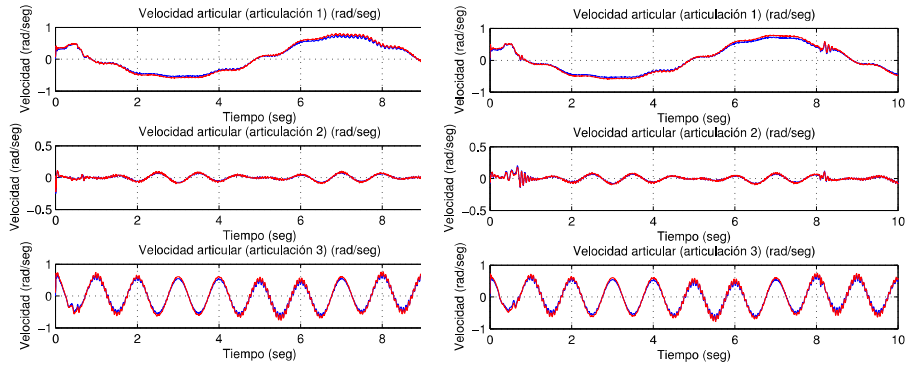


Figura 11. Velocidad articular control PD+G. **Figura 12.** Velocidad articular control par calculado.

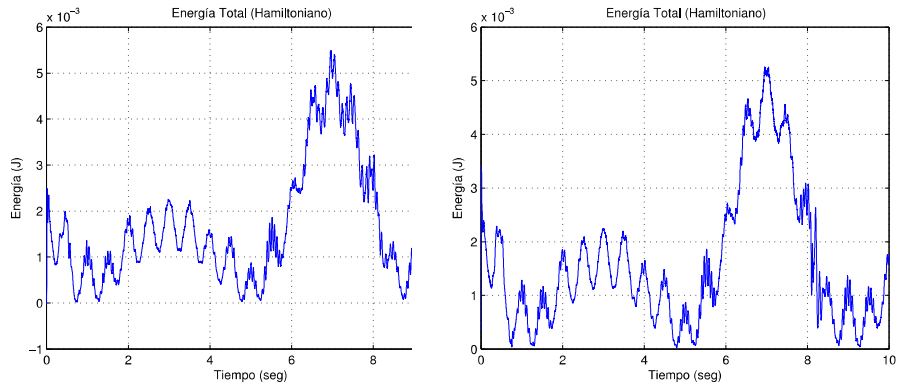


Figura 13. Energía total PD+G. **Figura 14.** Energía total par calculado.

5. Conclusiones

De los resultados obtenidos en este estudio comparativo de controles no lineales basados en pasividad, se puede visualizar que el control par calculado garantiza seguimiento de trayectorias y minimiza los errores de estado estable en posición y velocidad, sin embargo, es indiscutible que demanda mayor energía dada su cualidad de robustez.

Agradecimientos. Los primeros cuatro autores agradecen a conacyt, el primero a la beca conacyt número 490034, el segundo a la beca número 358028, el tercero a la beca número 410284 y el cuarto a la beca número 489658, y por supuesto al laboratorio de robótica avanzada e interfaces hápticas del citis por prestar su equipo e instalaciones para la realización de este trabajo experimental.

Referencias

1. Domínguez-Ramírez, Vicente Parra: Diseño e integración de una interfaz háptica activa realista. CINVESTAV, México (2004)
2. Ordaz-Oliver M. Oscar: Diseño e implementación de estrategias de control no lineal con propósitos de guiado hápticos. ITP, México (2009)
3. Ordaz Oliver J. Patricio, Domínguez Ramírez Omar A., Parra-Vega Vicente, Jarillo-Silva Alejandro: Diseño de estrategias de control basadas en pasividad para sistemas Euler-Lagrange aplicado en robots manipuladores. Capítulo 2. Mecatronica. IPN, México (2009)
4. Domínguez-Ramírez, V. López-Morales: Interfaces hápticas antecedentes y descripción. Universidad Autonoma del Estado de Hidalgo, Centro de Investigacion y de Estudios Avanzados del IPN (2006)
5. SensAble Technologies, PHANTOM premium 1.0/1,5A, 3D Touch Componets: HADware Instalacion and Technical Manueal Revision 6.5, 18 August, (2000).
6. Ortega Romeo, Loría Antonio, Nicklasson Per Johan and Sira-Ramírez Herbertt: Passivity-based control of Euler-Lagrange systems, Mechanical, electrical and electromechnical applications. Springer (1998).

Sintonización y optimización de un PID con algoritmos evolutivos

Armando Pedraza Martín, Omar Santos y Félix Castro

Universidad Autónoma del Estado de Hidalgo,
Pachuca, Hgo, México

pema09@hotmail.com, omarj@uaeh.edu.mx, fcastro@lsi.upc.edu

Resumen Actualmente la mayor parte de la industria utiliza un control automático sobre los sistemas de control de temperatura mediante un PID. Sin embargo, para que dicho control funcione adecuadamente es necesario que el PID sea sintonizado de manera eficaz para obtener los resultados deseados. El presente trabajo propone un algoritmo que logra encontrar los parámetros subóptimos de sintonización un PID que controla el proceso de una planta deshidratadora. La propuesta está basada en el algoritmo evolutivo *Hill Climbing* y tiene por objetivo encontrar los parámetros subóptimos de κ , τ y δ que son programados en un PID de corte industrial para el control de temperatura de la planta, reduciendo así el consumo de energía y minimizando el error de convergencia en menor tiempo.

Palabras clave: PID, algoritmos evolutivos, Hill climbing, Ziegler-Nichols.

1. Introducción

En los últimos años hemos sido testigos de un enorme esfuerzo mundial por detener el denominado cambio climático, uno de los mayores esfuerzos ha sido el reducir la cantidad de emisiones de gases del efecto invernadero, otra tendencia ha sido el reducir el consumo de energía, ejemplo de ello es la implantación del llamado horario de verano, sin embargo, es importante contribuir en todos los ámbitos para evitar el deterioro del planeta. En este contexto, es de suma importancia reducir el consumo de energía utilizada en las plantas productivas de las empresas, en primera instancia porque la empresa reduciría sus gastos, y en segunda para contribuir con el planeta. De hecho uno de los procesos industriales que más demandan energía, es el deshidratado de alimentos. Por tal razón, en el presente artículo se trata el problema de reducir el gasto de energía en una planta deshidratadora.

El problema abordado en el presente artículo es la optimización de 3 parámetros participantes en la sintonización de un control Proporcional, Integral y Derivativo (PID), utilizando un algoritmo evolutivo. El PID se utiliza en un sistema de control de temperatura, cuyo modelo matemático se aproxima mediante un sistema lineal de primer orden con retardo en la entrada. Dicho modelo fue propuesto para sintonización de PID's en [14] y es ampliamente usado actualmente en la industria para los

mismos propósitos [15]. Debido a que la mayor parte de las industrias hoy en día utilizan controladores PID para el control de diferentes procesos [1], la implementación de algoritmos evolutivos para la obtención de parámetros subóptimos se ha vuelto una técnica de suma importancia para optimizar diferentes plantas y procesos y prueba de ello son las referencias [2-8].

La importancia de sintonizar un controlador PID en un sistema de control de temperatura radica en optimizar la energía utilizada y el desempeño del proceso en general, ya que en este tipo de procesos el consumo de energía es muy alto. Por lo que al utilizar técnicas de optimización numérica se pretenden encontrar los valores subóptimos del controlador PID para el sistema de control de temperatura, lo cual implica minimizar el uso de energía, así como optimizar el funcionamiento de la planta. Para esto, se desarrolló un prototipo experimental de deshidratadora, el cual tiene el mismo modelo (primer orden con retardo) que una deshidratadora industrial. El prototipo experimental, cuenta con una fuente de aire, una fuente de calor y un pequeño túnel de viento. La entrada de control se realiza aplicando un voltaje a los ventiladores presentes en el sistema y la temperatura actúa como la salida, que a su vez es la variable del proceso. El voltaje aplicado a los ventiladores implica producir el aire necesario para mantener una temperatura sobre una referencia establecida. Se utiliza un controlador industrial PID industrial Honeywell DC1040 para el control del sistema, el cual se optimiza numéricamente en cuanto a su sintonización. Nuestra propuesta no implica la compra de hardware o software extra, solo se optimizan las ganancias del controlador y el resultado de dicha optimización se programa directamente en el controlador.

Basándose en estudios previos y experimentación en línea, este trabajo cuenta tanto con simulaciones como resultados experimentales en tiempo real, lo que permite tener mayor veracidad en la obtención de los resultados, así como observar el comportamiento real de la planta.

En la sección 2 se describe el sistema de control de temperatura de la planta tratada en el presente artículo, así como su modelo matemático. En el apartado 3, se da una breve descripción de los algoritmos evolutivos y en particular del Hill Climbing. La solución adoptada en el artículo se presenta en la sección 4, detallando la obtención de los parámetros que permiten la sintonización correcta del PID. En el apartado 5 se presentan los resultados obtenidos en los experimentos realizados y finalmente, las conclusiones y trabajos futuros se detallan en la sección 6.

2. Sistemas de control de temperatura

La función de un sistema de control de temperatura se logra mediante una combinación de subsistemas: actuador (en este caso los ventiladores), sensor (un termopar tipo J1), controlador (en este caso un PID) y planta (deshidratadora). Dentro del lazo de control, se mide la temperatura del aire en el túnel de viento de la deshidratadora (este aire caliente deshidrata al producto), el controlador compara la temperatura del aire (Variable del Proceso, o PV de sus siglas en inglés) y envía una señal de corrección al actuador (ventilador) para introducir la cantidad de flujo de calor o enfriamiento necesario para converger al punto de referencia establecido previamente por el

usuario. Sin embargo, existen varios factores que afectan la tarea del controlador, entre ellos están, el cambio de temperatura del medio ambiente, el tiempo de retraso, la ubicación de los componentes (ventiladores, sensor, controlador) y por supuesto, la selección correcta de los parámetros del PID. La regulación de la temperatura y el ancho de banda son los más afectados por estos factores [9]. En este trabajo nos centramos principalmente en optimizar la sintonización de los tres parámetros del controlador.

2.1. Obtención del modelo de la planta

A continuación se describe brevemente como está constituida la plataforma: el prototipo de deshidratadora consistente en una caja de acrílico (1) de 30 cm de cada lado con un túnel (2) como salida. Dentro de la caja se encuentra una rejilla eléctrica (3) que se utiliza como fuente de calor y se alimenta hasta con 120 Volts CA, pero sólo se introduce un voltaje fijo de 17.5 volts CD a través de una etapa de potencia (4). Los actuadores son 3 ventiladores (5) que tienen un rango de operación de 3V - 12 V CD. El sensor de temperatura (6) es un termopar tipo J1 y se coloca en el túnel y está conectado directamente al PID (7) para recibir la temperatura dada en ese momento. Cabe mencionar que la compensación e punta fría la realiza el PID industrial, solo se tiene que programar que tipo de termopar es con el que se cuenta. En la Fig. 1 se muestran los componentes del modelo de la planta correspondientes al número ilustrado en este párrafo.

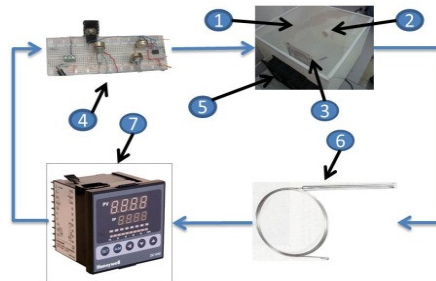


Fig. 1. Diagrama del prototipo de la planta deshidratadora

2.2. Modelo matemático

Para la obtención del modelo matemático se utilizó el método de Ziegler - Nichols en lazo abierto [14], dicho método es útil para determinar de manera experimental un modelo aproximado, mediante la respuesta de la planta en lazo abierto ante una entrada escalón unitario. La curva de respuesta se caracteriza por tres parámetros [14]:

- El tiempo de retardo, para este trabajo se emplea la variable h .
- La constante de tiempo T .
- La ganancia K del sistema.

De acuerdo con este método la planta se aproxima mediante:

$$\frac{Y(s)}{U(s)} = \frac{Ke^{-sh}}{Ts+1}. \quad (1)$$

Al utilizar este método, de manera experimental se encuentran los valores de los parámetros h , T y K , para el sistema. La temperatura externa al momento de obtener la respuesta del sistema fue de 25 °C. Utilizando el método Ziegler y Nichols se obtiene:

$$\frac{Y(s)}{U(s)} = G(s) = \frac{0.1e^{-14s}}{226s+1}. \quad (2)$$

La ecuación (1) describe un sistema lineal de primer orden con retardo en la entrada, donde $Y(s)$ es el mapeo de la temperatura y $U(s)$ es el voltaje aplicado a los ventiladores [10]. La Fig. 2 muestra la simulación de la planta (2) ante el escalón unitario, la cual es muy similar a la obtenida experimentalmente.

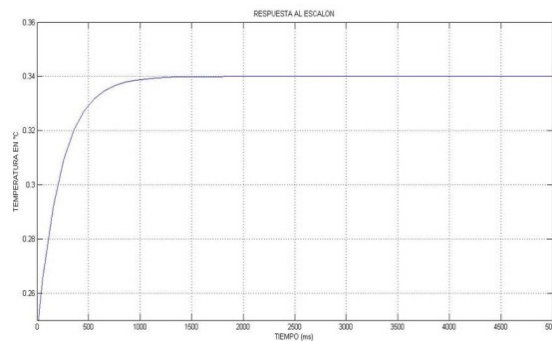


Fig. 2. Respuesta al escalón del modelo de la planta (2)

3. Algoritmos evolutivos

La idea de los algoritmos evolutivos comienza a partir de la resolución de un problema específico, por tanto se va a convertir en un proceso de búsqueda de una solución, llamado comúnmente "espacio de búsqueda". Cuando se considera la búsqueda de la mejor solución posible, se convierte en un problema de optimización. Cuando los espacios de búsqueda son pequeños o simples se pueden utilizar métodos de optimización y búsqueda clásicos como lo son métodos de búsqueda de tipo directo [11], sin embargo, éstos presentan algunas limitaciones:

1. La convergencia a una solución óptima depende de la condición inicial de la que parte la búsqueda.
2. Muchos algoritmos son propensos a verse atrapados en un óptimo local.
3. Falta de robustez, donde, un algoritmo eficaz en la resolución de un determinado problema puede no ser adecuado para otro tipo diferente de problema.
4. Algunos algoritmos no son capaces de tratar adecuadamente problemas con variables discretas.

3.1. Algoritmo de Hill Climbing

Como se mencionó, en el presente trabajo se adoptó como método de optimización el algoritmo Hill Climbing ("ascenso de colinas") por lo que a continuación se describe brevemente su funcionamiento.

Hill Climbing es un algoritmo que iterativamente permite ir obteniendo una mejor solución. Comienza con un punto en el espacio de búsqueda, una vez seleccionado este punto, si es el mejor se convierte en el punto actual, si no, es seleccionado otro punto vecino a partir del actual y es evaluado y así sucesivamente hasta encontrar la solución buscada o hasta que se satisfaga un criterio de paro. Debido a que sólo permite ir mejorando la solución, la cadena se muta, y si esta mutación proporciona un mejor punto que el actual, éste se reemplaza. El método termina cuando se encuentra un mejor punto y el actual no puede obtener mejorías o en todo caso termina cuando se alcanza el número definido de iteraciones representativamente [12].

El algoritmo Hill Climbing se considera un algoritmo voraz, debido a que es un algoritmo iterativo, donde en cada iteración se selecciona el mejor resultado para así obtener un resultado global. A este algoritmo se le considera local debido a que a partir del mejor punto actual busca únicamente en las instancias inmediatas (vecinos). De igual forma se considera una estrategia irrevocable ya que no permite regresar a otra alternativa encontrada con anterioridad [12]. De los elementos utilizados en un algoritmo evolutivo, Hill Climbing no utiliza la opción de cruce, ya que su funcionamiento se basa sólo en la mutación de un valor inmediato elegido a partir de la mejor solución actual y así evaluar cuál es la mejor solución. La Fig. 3 ilustra un diagrama con el funcionamiento general del algoritmo Hill Climbing.

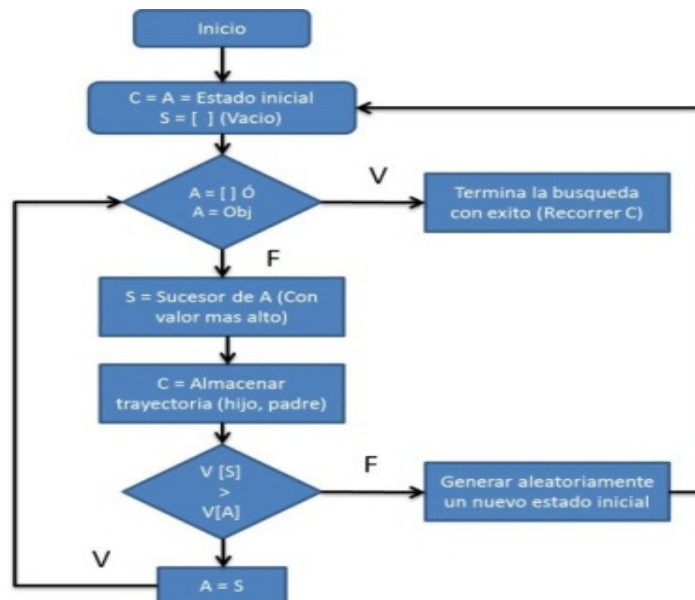


Fig. 3. Diagrama de flujo del algoritmo Hill Climbing

4. Optimización numérica del control

El objetivo principal de la optimización es encontrar los valores de los parámetros del control, los cuales dependen del sistema a controlar. En efecto, la optimización de dichos parámetros así como sus restricciones, dependen del modelo matemático de la planta, al obtener este modelo se puede utilizar un algoritmo de optimización para encontrar la solución.

Una vez elegido el algoritmo se aplica al modelo matemático en lazo cerrado con el controlador con los parámetros propuestos en cada paso del algoritmo numérico de búsqueda, aquí el usuario debe de ser capaz de reconocer si se tuvo éxito en la búsqueda de la solución. Una vez obtenidos los resultados se analizan con las condiciones de optimalidad para comprobar si el conjunto de variables es la solución al problema. Si el modelo sufre algún cambio en sus variables, el problema de optimización se debe resolver de nuevo y se repite el proceso [13].

4.1. Algoritmo Hill Climbing para la obtención de los parámetros kp , ki y kd

Como ya se mencionó, Hill Climbing al no ser un algoritmo exhaustivo sólo permite encontrar una solución, en el problema abordado en el presente artículo es necesario encontrar tres parámetros, por tanto el primer valor encontrado para el controlador se genera a partir de la función de optimización; sin embargo, el segundo valor depende del resultado encontrado en el primer ciclo Hill Climbing, y para el tercero depende tanto del primero como del segundo valor encontrado, esto dentro de las funciones de optimización, por ello se realizan tres ciclos Hill Climbing, uno para kp , otro para ki y uno más kd . Esto de acuerdo con el algoritmo general del PID:

$$u(t) = Kp e(t) + Ki \int_0^t e(t) dt + Kd e(t),$$

donde $e(t)$ = Referencia-Variable del Proceso. La Fig. 4 representa el diagrama de flujo del algoritmo Hill Climbing que se utiliza para describir de manera general el proceso de obtención de los tres parámetros kp , ki y kd . Posteriormente, en las subsecciones siguientes se describen los pasos realizados para la obtención de los parámetros mencionados.

4.2. Ciclo Hill Climbing para kp

A continuación se describe el proceso para la obtención del parámetro kp .

- (a) El primer paso es inicializar las variables kp , ki y kd
- (b) El siguiente paso es iniciar el primer ciclo *for*
- (c) Se simula la planta en lazo cerrado donde se obtiene el primer valor subóptimo para kp donde el primer resultado es cero, sin embargo, cuando el valor sea mutado por el algoritmo irá cambiando dependiendo si encuentra o no una mejor solución.
- (d) Posteriormente se calcula el índice de desempeño.
- (e) A continuación se inicia un segundo ciclo *for* anidado dentro del primero.
- (f) Se realiza una mutación con un 50 % de probabilidad de cambio.

- (g) Una vez realizada la mutación se simula nuevamente la planta en lazo cerrado donde se obtiene el valor subóptimo para k_p mutado.
- (h) Se calcula el índice de desempeño para este nuevo resultado.
- (i) Ahora se comparan los resultados obtenidos entre el primer resultado obtenido y el resultado de la mutación.
- (j) Si no se encuentra una mejor solución se realiza nuevamente la mutación, en caso contrario, donde se encuentre una mejor solución se actualiza el resultado al valor subóptimo encontrado.

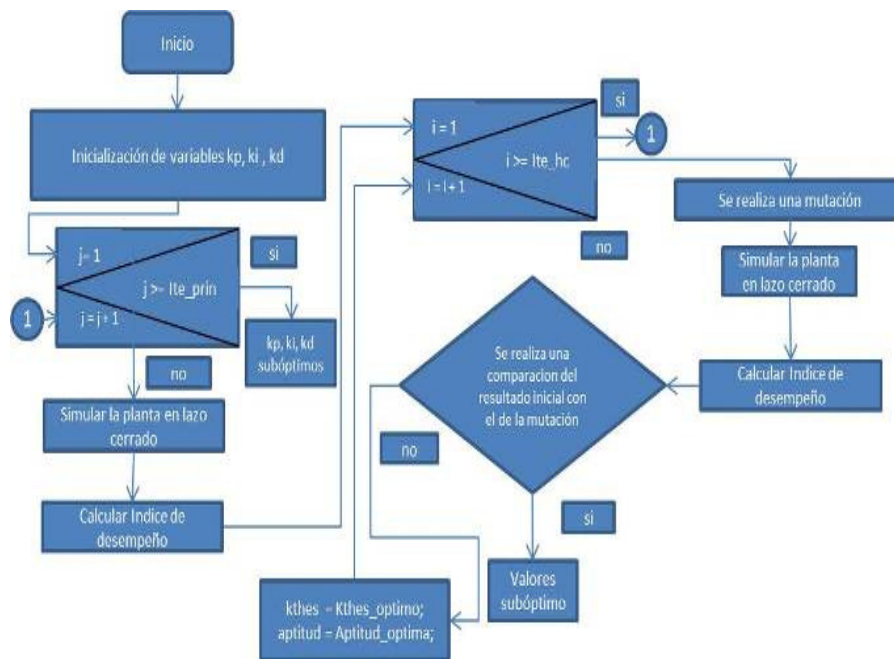


Fig. 4. Diagrama de flujo Hill Climbing

4.3. Ciclo Hill Climbing para k_i

Una vez encontrado el resultado subóptimo de k_p se realiza nuevamente el procedimiento realizado para la obtención de k_p pero ahora para k_i .

- (a) Se inicia nuevamente el primer ciclo *for* ahora para k_i .
- (b) Se simula la planta en lazo cerrado, donde, se soluciona la ecuación (2) mediante el método Ziegler – Nichols, con este resultado se crea un espacio de estado del modelo y se obtiene el lazo abierto, que a su vez es convertido a lazo cerrado, donde se obtiene la respuesta de la temperatura y el voltaje aplicado a los ventiladores.
- (c) Se calcula el índice de desempeño.

- (d) En este punto se repiten los pasos (e) y (f) realizados en el ciclo Hill Climbing para kp .
- (e) Para obtener el resultado se repite el paso (b) realizado en este ciclo Hill Climbing para ki , pero para el valor mutado.
- (f) Y por último se repiten los pasos (h), (i) y (j) realizados en el ciclo Hill Climbing para kp .
- (g) Se actualiza el valor ki subóptimo encontrado.

4.4. Ciclo Hill Climbing para kd

Una vez encontrado el resultado subóptimo de kp y ki , se realiza nuevamente el procedimiento realizado para la obtención de kp pero ahora para kd .

- (a) Se inicia nuevamente el primer ciclo *for* ahora para kd .
- (b) Se simula la planta en lazo cerrado, donde, se va a realizar una discretización, esto quiere decir, convertir una señal analógica a digital, debido a que el resultado es una gráfica senoidal, se utilizan ecuaciones en diferencias para segmentar la gráfica senoidal e ir encontrando la solución.
- (c) Se calcula el índice de desempeño.
- (d) En este punto se repiten los pasos (e) y (f) realizados en el ciclo Hill Climbing para kp .
- (e) Para obtener el resultado se repite el paso (b) realizado en este ciclo Hill Climbing para kd , pero para el valor mutado.
- (f) Y por último se repiten los pasos (h), (i) y (j) realizados en el ciclo Hill Climbing para kp .
- (g) Se actualiza el valor kd subóptimo encontrado.

5. Resultados experimentales de control

Para culminar este trabajo se presentan los resultados experimentales obtenidos y se presenta una comparación con el método Ziegler – Nichols [14], para ello se muestran de manera gráfica los resultados.

5.1. Temperatura obtenida con Ziegler - Nichols y el método propuesto

La temperatura inicial dentro de la cámara fue de 29 °C, el valor de referencia fijado fue de 50 °C, la temperatura externa fue de 23.2 °C. Los resultados de la temperatura obtenidos con el método Ziegler - Nichols, se muestran en la Fig. 5.

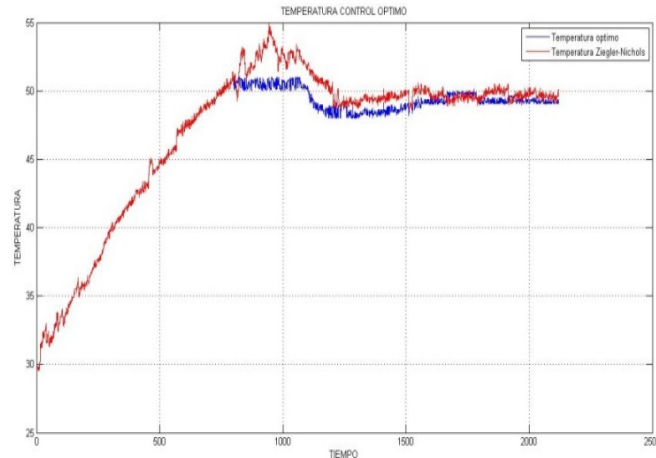


Fig. 5. Temperatura obtenida con Ziegler – Nichols y Hill Climbing

Se observa en la Fig. 5 que la temperatura se logra estabilizar con el método Ziegler – Nichols, sin embargo, una vez que se alcanza el “set point” fijado a 50 °C se llega a elevar la temperatura a los 55 °C. Los resultados obtenidos con el método propuesto estabilizan la temperatura a 50 °C, evitando sobre picos que desestabilicen la planta. Sin embargo, en un momento ésta disminuye a 49 °C, sin embargo, mantiene la estabilidad cercana a los 50 °C.

5.2. Voltaje obtenido con Z - N y el método propuesto

La Fig. 6 representa el voltaje aplicado a los ventiladores con el método Ziegler – Nichols y con el propuesto en el presente artículo.

Se observa que con el método Ziegler – Nichols el voltaje aplicado a los ventiladores se eleva hasta los 8.5 volts y logra disminuir a los cercanos 6.4 volts. Con el método propuesto se aplica un voltaje en su máximo punto de 6.4 volts, pero logra estabilizar un voltaje mínimo acercándose a los 4.5 volts. Por lo que el gasto de energía con el método propuesto es menor en comparación con Ziegler – Nichols.

Por último la Tabla 1 muestra un comparativo de los resultados obtenidos, donde se pueden observar los siguientes resultados:

- 1.- El método propuesto garantiza estabilizar la temperatura en un menor tiempo, con lo que se evita que la temperatura se pueda elevar.
- 2.- El método propuesto disminuye considerablemente el consumo de energía que, en comparación con el método Ziegler-Nichols se minimiza en 2 volts.
- 3.- Con el método propuesto se logra una convergencia del error a cero en un menor tiempo en comparación a Ziegler-Nichols.

De esta forma se demuestra que el método Ziegler-Nichols logra una estabilización en la planta; sin embargo, el método propuesto lo realiza optimizando aún más los resultados lo que indica la eficiencia del método propuesto.

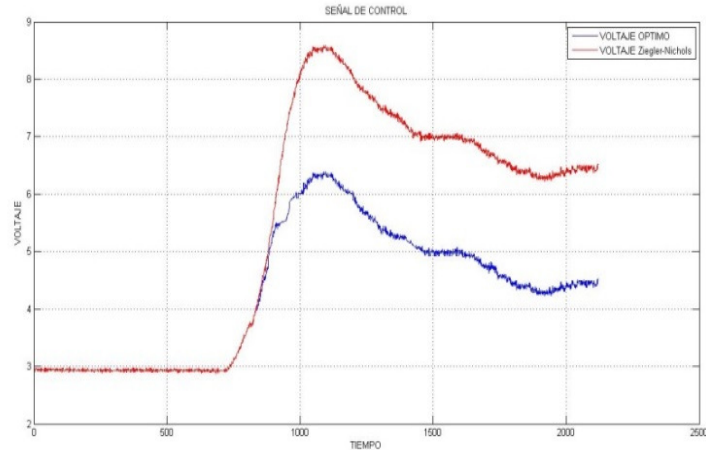


Fig. 6. Voltaje aplicado a los ventiladores con ambos métodos

Tabla 1. Resultados comparativos

	Máximo sobreimpulso	Tiempo de convergencia	Tiempo pico	Tiempo de crecimiento	Voltaje máximo aplicado
Ziegler - Nichols	5 °C	1200 seg	950 seg.	700 seg.	8.5 VCD
Método Propuesto	1 °C	700 seg	751 seg.	700 seg.	6.4 VCD

6. Conclusiones

Los resultados con el método Ziegler – Nichols y con el propuesto en el presente trabajo de investigación parecieran ser similares, sin embargo, la diferencia radica principalmente en minimizar la energía y la convergencia del error a cero, donde el método propuesto lo realiza en menor tiempo y demuestra ser subóptimo, además el método empleado, no involucra software o hardware extra en el lazo de control.

La principal aportación de este trabajo, es la metodología que permite obtener los parámetros subóptimos para un PID para poder optimizar el tiempo de respuesta y energía. Como trabajos futuros se pretende realizar experimentos en plantas más grandes, donde el gasto de energía y el tiempo de convergencia sean mayores, permitiendo una evaluación más realista de la aportación del método propuesto, así mismo se pretende comparar los resultados con otros métodos de optimización.

Referencias.

1. K J. Aström ant T. Hagglund: PID Controllers: Theory, Design and Tuning., Instrument Society of America (1995)

2. Mohammed Obaid Ali, S. P. Koh, K. H. Chong, S.K.Tiong and Zeyad Assi Obaid: Genetic Algorithm Tuning Based PID Controller for Liquid-Level Tank System. Department of Electronics and Communication Engineering, Universiti Tenaga Nasional, 11-13 October (2009)
3. Neenu Thomas, Dr. P. Poongodi: Position Control of DC Motor Using Genetic Algorithm Based PID Controller. In: Proceedings of the World Congress on Engineering 2009 Vol II, WCE 2009, London, U.K., July 1-3 (2009)
4. Musa Usman Otaru, Naji A. Al-Musabi, Samir A. Al-Baiyat: Robust PID Stabilizer Design Using Genetic Algorithms. Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudia Arabia.
5. S.Ravi, P.A.Balakrishnan: Genetic Algorithm Based Temperature Controller For Plastic Extrusion System. Department of EEE, Nandha Engineering College, Tamilnadu, India.
6. S.M.Girirajkumar, Atal.A.Kumar,N.Anantharaman: Speed Control of a Real Time D.C. Shunt Motor Using SA Based Tuning of a PID Controller. International Journal of Computer Applications (0975 .8887), Volume 5.No.11, August (2010)
7. G. Sánchez-Díaz, O. Santos-Sánchez: Suboptimal control based on hill-climbing method for time delay system, IET Control Theory Appl. 1 1441-1450 (2007)
8. A. Visioli: Optimal tuning of PID controllers for integral and unstable processes, IEEE Proceedings online no. 20010197 -Control Thieov Appl , Vol 148, No 2, March (2001)
9. Dale Patrick and Stephen W. Fardo, Industrial Process Control Systems, Delmar Publishers.
10. L. Rodríguez Guerrero: Sistema de optimización y supervisión desarrollado en Java. Caso de estudio: Proceso de temperatura con retardo, Tesis de la maestría en Automatización y Control, Universidad Autónoma del Estado de Hidalgo, Pachuca Hgo., Diciembre (2010)
11. K. Deb: Optimization for Engineering Design: Algorithms and Examples, Prentice-Hall (1995)
12. B. López Takeyas: Algoritmo Hill Climbing, Instituto Tecnológico de Nuevo Laredo, Ingeniería en Sistemas Computacionales, Octubre (2005)
13. J. Nocedal and Stephen J. Wright: Numerical Optimization, Second Edition, Springer
14. J. G. Ziegler and N. B. Nichols, Optimum Settings for Automatic Controllers, The annual meeting, New York, December 1-5 (1941)
15. Página del software industrial *Expert Tune*, <http://www.expertune.com/>, consultada en diciembre del 2012.

Lógica y ontologías

Comportamientos en simulación de multitudes: revisión del estado del arte

Oriam De Gyves¹, Leonel Toledo¹ y Isaac Rudomín²

¹ Tecnológico de Monterrey, Campus Estado de México,
México

² Barcelona Supercomputing Center,
Barcelona, España

Resumen Las grandes aglomeraciones de peatones son un fenómeno muy común en la actualidad y han sido estudiadas por diferentes investigadores. La simulación de multitudes es un tema de investigación que surge para poder reproducir este fenómeno y poder aprovecharlo para diferentes aplicaciones. En este trabajo se identifican los problemas que existen en la simulación de multitudes en tiempo real, específicamente los relacionados con los comportamientos de los agentes virtuales, y cómo éstos han sido abordados por la comunidad científica. El propósito de este artículo es el de proveer a los investigadores con información relevante y del estado del arte en la simulación de multitudes.

Palabras clave: modelación, comportamiento de multitudes.

1. Introducción

El uso de aplicaciones para la simulación de multitudes está creciendo rápidamente [1], de la industria del entretenimiento a la planificación urbana y la seguridad pública, los gobiernos y las industrias privadas cada vez están más interesados en los posibles usos de esta área de investigación.

Películas como *El Señor de los Anillos*, *Avatar* y *Guerra Mundial Z* hacen uso extensivo de técnicas desarrolladas en la simulación de multitudes, como se puede ver en la Figura 1. En las escenas de acción de gran presupuesto de estas películas, miles de personajes complejos interactúan entre si y con su entorno, logrando una sensación de inmersión mas profunda en los espectadores. De igual manera, la simulación de multitudes también se ha utilizado en la industria de los videojuegos con el fin de lograr escenas más realistas y llenas de vida [3–5].

Gobiernos alrededor del mundo están tratando de utilizar este tipo de simulaciones para planificar sus ciudades, asegurando que la infraestructura sea suficiente para las necesidades de la población. Prototipos de nuevos avances tecnológicos, como las ciudades inteligentes donde coches autónomos interactúan con los peatones en un ambiente seguro son desarrollados y probados utilizando aplicaciones para la simulación de multitudes [6].

Este tipo de tecnología también es usada para la seguridad pública, en lugares y situaciones donde densos grupos de personas se ven involucradas, como se



Figura 1. Escena de la película *Guerra Mundial Z* con un uso intensivo de simulación de multitudes. Referencia original de [2].

observa en la Figura 2. El gobierno puede hacer uso de la simulación de multitudes con el fin de prevenir situaciones de desastre en actos públicos, aeropuertos y pasos subterráneos [8–14].

El objetivo principal en el estudio y simulación de multitudes es representar con precisión grupos de personajes autónomos, llamados agentes virtuales, con reglas y entornos lo más cercanos a aquellos encontrados en la vida real. Entre los objetivos también se encuentra el generar algoritmos optimizados capaces de producir visualizaciones en tiempo real, lo que significa que la frecuencia de fotogramas debe ser de al menos 30Hz.

El número de personajes que constituyen una multitud varía en los diferentes trabajos de investigación, sin embargo, los investigadores a menudo tratan de aumentar el número tanto como sea posible, desde unos pocos cientos hasta millones, manteniendo la meta de visualizaciones en tiempo real [15].

A pesar de que el problema de la simulación de multitudes lo han estudiado especialistas en gráficas computacionales, muchas áreas se han sumado para crear simulaciones cada vez más complejas. Físicos y especialistas en Inteligencia Artificial (IA), entre otros, han creado modelos para simular grandes cantidades de agentes virtuales.

En las páginas siguientes, varios artículos y algoritmos para simular comportamientos realistas para multitudes se presentan, desde las primeras simulaciones de Reynolds con agentes reactivos, hasta agentes capaces de formar lazos sociales.



Figura 2. Multitud de peatones en lugares públicos. Referencia original de [7].

2. Trabajo existente

Diversidad visual, animación de personajes, cantidad de personajes, inteligencia artificial y la planificación de rutas son problemas comunes que los investigadores tratan de resolver en las simulaciones de multitudes [16].

El área de la simulación de multitudes ha sido estudiada por investigadores a niveles microscópico [17] y macroscópico [18]. Las simulaciones macroscópicas están enfocadas en el movimiento realista de la multitud como un todo, por otro lado, las simulaciones microscópicas se centran en el movimiento real de los individuos [19].

Comúnmente, los grupos de investigación crean sistemas propios para sus continuos avances en la simulación de multitudes [16, 20–23]. En las siguientes páginas se presentan un conjunto de trabajos destacados en el área.

2.1. Comportamientos de dirección

A pesar de que su investigación no se centra en peatones, Craig Reynolds presentó uno de los primeros trabajos de investigación que se centró en los grupos de agentes virtuales autónomos y sus comportamientos en 1987 [24]. La principal inspiración de Reynolds son las bandadas de aves ya que, como el autor describe, el concepto es simple pero visualmente complejo.

Según las investigaciones de Reynolds, un pájaro es consciente de tres elementos durante el vuelo, si mismo, dos o tres vecinos cercanos, y el resto de la bandada. Estos son importantes porque permiten a las aves formar parte de un grupo, independientemente del número de aves en el mismo.

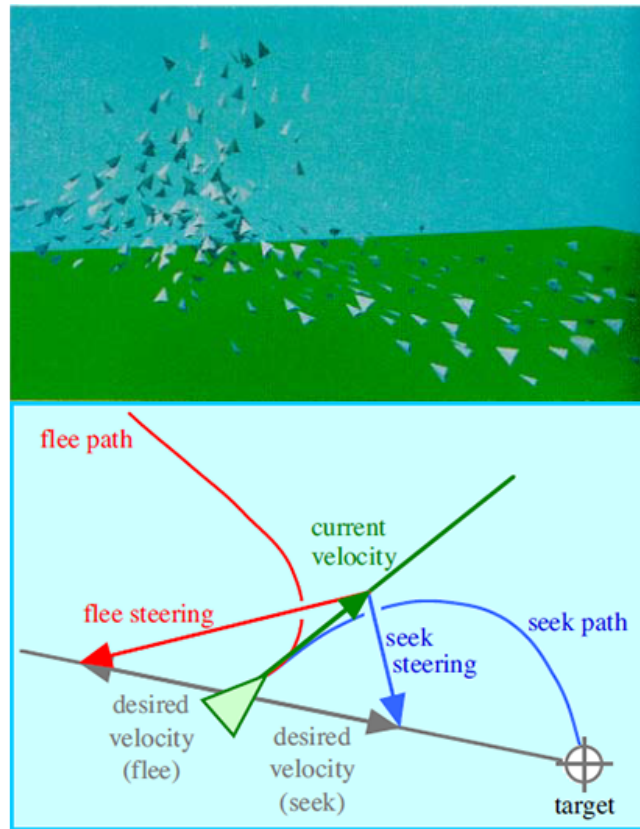


Figura 3. La imagen de arriba representa el resultado de los primeros *boids*, referencia original de [24]. La imagen de abajo muestra los comportamientos *seguir* y *huir*, referencia original de [25].

Para simular una bandada de aves, Reynolds hace uso de tres reglas simples, *separación*, *alineación* y *cohesión*, por orden de importancia. Reynolds acuña el término *boids*, para referirse a los agentes que siguen sus reglas de movimiento.

En 1999 Reynolds publica los avances de su investigación, donde describe reglas adicionales de navegación para los agentes autónomos en un mundo virtual [25]. El objetivo es que los agentes se muevan *de manera vivaz e improvisada*. Nombra estas reglas como comportamientos de dirección (*steering behaviors* en inglés). Los comportamientos presentados en estos trabajos son descritos como

bloques de construcción, que se pueden combinar con el fin de obtener patrones más complejos de comportamientos. Los resultados de los trabajos de Reynolds pueden verse en la Figura 3.

Se considera un comportamiento reactivo a todas aquellas acciones que son desencadenadas mediante un evento, es decir mientras no se registre ningún estímulo del ambiente o incluso de algún otro agente involucrado en la simulación, los agentes que pertenezcan a la categoría reactiva no tienen ninguna razón para reaccionar.

Millan et al. genera máquinas de estado finitas desde archivos XML y las guarda en imágenes que los agentes pueden consultar [15]. Pelechano y Badler han combinado reglas de percepción y comportamientos reactivos para dirigir agentes en entornos virtuales [26]. Aunque simples los comportamientos reactivos, si son implementados correctamente pueden tener como consecuencia comportamientos emergentes que pueden generar simulaciones más realistas.

Una simulación basada en reglas que no requieren razonamiento complejo debido a que el medio ambiente es el responsable de seleccionar la mejor acción de los agentes a seguir en diferentes situaciones es presentada en 2011 [19]. Se han combinado predicciones de espacio-tiempo, comportamientos reactivos, y movimientos de dirección en plataformas dedicadas a la simulación de multitudes [23].

2.2. Búsquedas de proximidad

Para simular cómo los humanos resuelven las colisiones con mayor precisión, cada agente tiene que estar al tanto de un entorno dinámico y adaptar su trayectoria continuamente. En escenarios reales, los peatones dentro de una multitud tienen que encontrar su camino a través de otros individuos. En simulaciones de multitudes, los agentes deben ser conscientes de obstáculos dinámicos y otros agentes en el entorno.

Un algoritmo ingenuo de búsqueda de vecinos tiene una complejidad de $O(n^2)$, donde n es el número de agentes. Esto significa que cada agente tiene que consultar a todos los otros agentes por su posición y otra información importante. Estructuras de datos especiales surgieron para tratar de resolver este problema, reducir la complejidad de las búsquedas de proximidad y permitir simulaciones de grandes multitudes.

En las estructuras jerárquicas, como los *octrees*, el espacio se subdivide en varias regiones que contienen agentes [27]. Algunos autores utilizan árboles *kd* para que consultar a los vecinos más cercanos sea un proceso más eficiente [28], [29]. Hay implementaciones en paralelo de árboles *kd* tanto en la unidad de procesamiento central (CPU) [30] como en el procesador gráfico (GPU) [31].

Bleiweiss presentó una implementación en paralelo de una biblioteca popular para evasión de colisiones, obteniendo un aumento de velocidad de 4.8X en comparación con la implementación original [32]. En la versión paralela, los autores también cambiaron el método de búsquedas de proximidad de un árbol *kd* a un método basado en hash con el fin de mejorar el rendimiento en el GPU.

Una estructura de datos topológica de resolución múltiple para la búsqueda de vecinos, que codifica la información para cada agente en una estructura celular fue presentado por Jund et al. [33]. Cada agente consulta sólo las celdas adyacentes en la búsqueda de sus vecinos, lo que reduce el número de cálculos de distancia. Passos et al. almacena a los agentes dentro de una matriz ordenada y usa una técnica de vecindario extendido de Moore (*extended Moore neighborhood* en inglés) para encontrar vecinos [34].

Manocha et al. presentó una técnica que se basa tanto en la localización geométrica como en consultas de proximidad basadas en imágenes para encontrar vecinos [35]. Los autores toman muestras de la geometría, hacen la rasterización de los objetos de interés y tratan los píxeles sobrescritos como intersecciones. Millan et al. propuso una técnica basada en imágenes en la que cada agente pinta un área de influencia en su posición de que otros agentes leen para moverse a través del entorno [15].

Mediante el uso de diagramas de Voronoi, el grupo GAMMA calcula mapas de distancia de superficie para cada agente, que codifica la distancia con los vecinos más cercanos [36]. Estos mapas se utilizan tanto para las búsquedas de proximidad como en la planificación de rutas. El mismo grupo de investigación ha utilizado diagramas de Voronoi de primero y segundo orden para las búsquedas de proximidad en otros trabajos [37,38].

2.3. Evasión de colisiones y planeación de rutas

La planeación de rutas tiene el objetivo de obtener el mejor camino desde un punto de origen a un destino. Cuando el ambiente tiene obstáculos dinámicos se requiere además de un sistema que evalúe constantemente nuevos caminos libres de colisión. Contrario a la evasión de colisiones, comúnmente la planeación de rutas no se realiza en cada paso de la simulación.

Chenney propone una técnica para la representación y el diseño de campos de velocidad usando autómatas celulares en 2004 [39]. Los campos de velocidad creados usando esta técnica son útiles para crear movimientos de flujo, que son seguidos por los peatones con el fin de moverse a través de un entorno. En 2011 Zhang et al. presenta un modelo en el que las celdas de un autómata celular representan posiciones discretas en el espacio, utilizadas por los peatones para moverse al cambiar de una celda a otra [40].

Hubert hace uso de autómatas celulares en la simulación de grandes eventos [41]. En este modelo, las celdas también son una representación discreta del espacio geográfico en el que los agentes son capaces de moverse, aunque representan un área en lugar de ranuras disponibles, por lo que los agentes pueden moverse a través de ellas, no solo de una a otra.

Treuille y Popović presentan el modelo *Continuum Crowds* en 2006, en el cual unifican la planeación de rutas y la evasión de colisiones utilizando campos potenciales y de velocidad para guiar a los agentes virtuales [42]. En este modelo se logran conseguir simulaciones en tiempo real, movimiento uniforme y comportamientos emergentes.

En 2009 Fischer et al. utiliza campos potenciales para permitir que los agentes negocien el espacio, evitar colisiones y alcanzar sus metas [43]. El algoritmo es una versión paralela de una técnica anterior de los mismos autores [44]. La aplicación se realizó en el GPU utilizando la plataforma *Compute Unified Device Architecture* (CUDA) de Nvidia, teniendo cuidado especial para reducir el número de transacciones de memoria entre CPU y GPU.

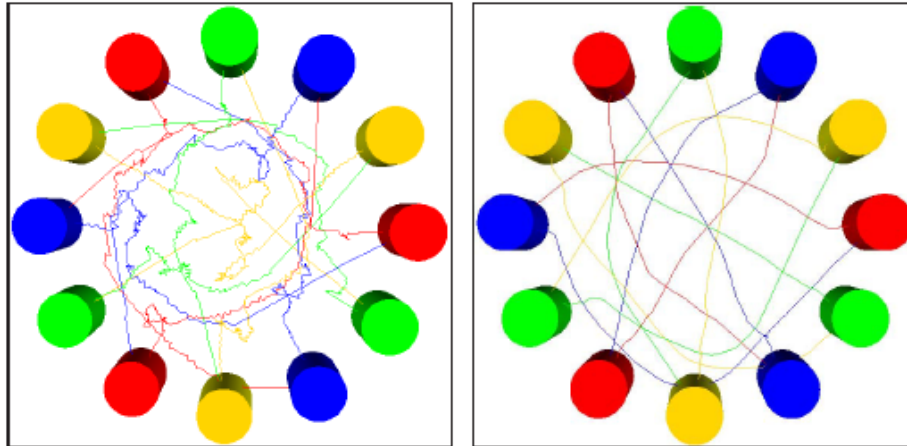


Figura 4. A la izquierda una simulación utilizando el algoritmo VO. A la derecha el algoritmo RVO. Referencia original de [28].

Guy et al. presenta un método que calcula trayectorias, reduciendo al mínimo el esfuerzo que los agentes necesitan llevar a cabo con el fin de llegar a su destino [29]. El algoritmo es capaz de evitar colisiones con otros agentes y los obstáculos, al tiempo que permite simulaciones en tiempo real.

Fiorini y Shiller introducen el término *Velocity Obstacles* (VO), una técnica geométrica para detectar objetos que chocan en el futuro con sus velocidades, asumiendo esas velocidades no cambian [45]. VO es el conjunto de todas las velocidades que conducirán eventualmente a dos agentes a una colisión, por lo que éstos deben elegir velocidades que no sean elementos del conjunto. El algoritmo genera caminos libres de colisión, pero con un gran número de oscilaciones, que hacen al movimiento poco natural.

En 2008 van den Berg et al. amplía la idea previa en *Reciprocal Velocity Obstacles* (RVO) [28]. Hay dos adiciones principales a la algoritmo de Fiorini; las oscilaciones en las rutas se eliminan mediante el supuesto que otros agentes se están dirigiendo utilizando las mismas reglas y dividiendo el esfuerzo entre todos los agentes involucrados como puede observarse en la Figura 4; además de un método para indicar la cantidad que un agente debe desviarse de su trayectoria original cuando detecta otros agentes en su camino. De esta manera crean las prioridades de movimiento para la simulación de agentes con diferentes rangos.

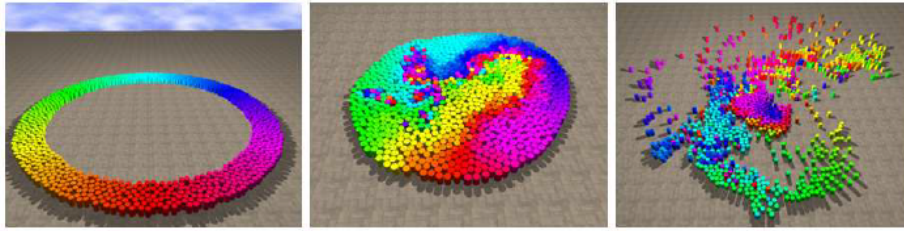


Figura 5. En la prueba, los agentes tienen que moverse a la posición opuesta de donde comienzan en el círculo. Referencia original de [46].

El mismo grupo de investigación de van den Berg amplía aún más el algoritmo de RVO, reduciendo el problema a un programa lineal de baja dimensión [46]. Mediante el uso de este algoritmo, los autores fueron capaces de simular miles de personajes, aún en CPU. En la Figura 5 se observa una prueba con 1,000 agentes en tiempo real.

Thalmann et al. presenta un sistema de navegación interactiva para multitudes en 2013 [47]. Su sistema permite a un usuario seleccionar un *avatar* de entre la multitud y controlar su movimiento a través del entorno virtual, por medio de una interfaz natural que captura los movimientos del usuario.

2.4. Comportamientos sociales

En 1995 Helbing y Molnár presentaron el modelo de fuerzas sociales, el cual se refiere a aquellas fuerzas que no se obtienen directamente del ambiente en el que se encuentran los peatones, sino que representan sus motivaciones internas para realizar determinadas acciones, tomando en cuenta algunos estímulos del medio [48]. El proceso psicológico y mental es la etapa en la que se evalúan las entradas del medio ambiente para que los agentes puedan tomar una decisión. La velocidad de los peatones se modifica para reflejar las decisiones tomadas durante esa etapa.

La investigación señala que existen dos clasificaciones principales de los estímulos, situaciones simples o conocidas y situaciones complejas o nuevas. El primer tipo de estímulo puede ser modelado por medio de fuerzas sociales, mientras que para el segundo los autores sugieren el uso de modelos probabilísticos. Aunque este modelo se aplica mejor a situaciones simples o previamente conocidas por los peatones, se afirma que los peatones normalmente se enfrentan a situaciones a las que ya están acostumbrados.

Helbing et al. estudia las diferencias entre las situaciones normales y de pánico para las simulaciones de multitudes [17]. En situaciones normales, los peatones tratan de seguir la ruta de menor esfuerzo hacia su destino e intentan mantener una velocidad deseada a lo largo del recorrido, además de conservar un espacio entre ellos y otros peatones y obstáculos. Por otro lado, en situaciones de pánico y a medida que aumenta el nerviosismo, los peatones tratan de moverse mucho

más rápido y la distancia entre las personas se reduce hasta el punto en que los peatones comienzan a empujarse unos a otros. Estas reacciones conducen a una falta de coordinación, donde el escape se ralentiza por cuellos de botella. Otra diferencia importante es que las personas tienden a mostrar un comportamiento de manada en situaciones de pánico, que es cuando la gente hace lo que hacen los demás, por lo general haciendo caso omiso de las rutas y salidas alternativas.

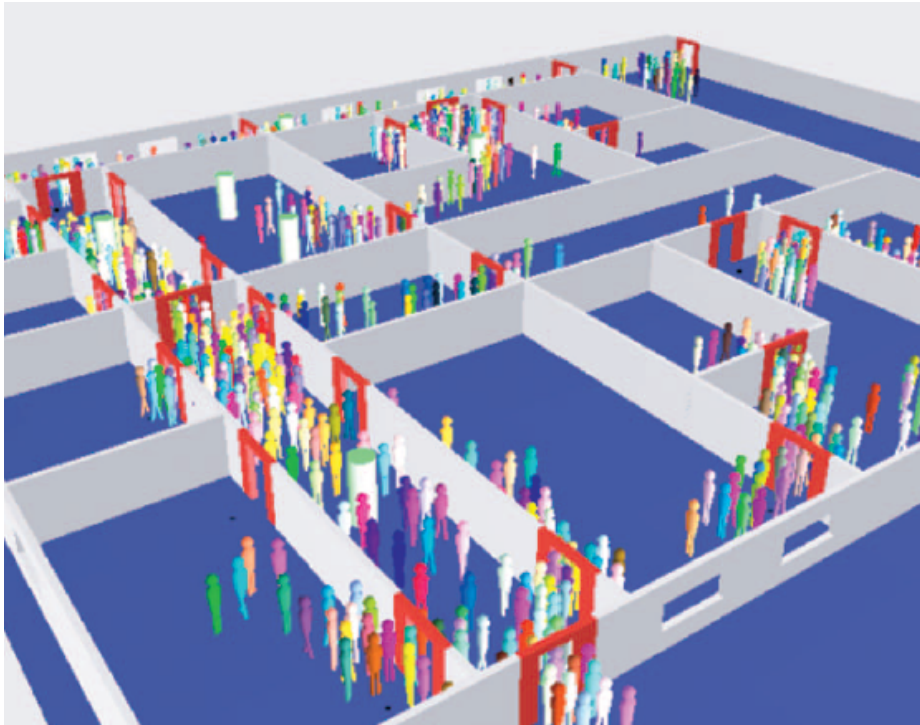


Figura 6. Líderes guían a otros agentes a las salidas en simulaciones de desastres. Referencia original de [49].

Una simulación de multitudes basada en teorías psicológicas es propuesta por Beltaief et al. en 2011 [50]. Los factores psicológicos que consideran importantes para tener simulaciones precisas son el psico-físico (percepción y velocidad) y psico-social (objetivos y preferencias). Usando los dos factores psicológicos, los autores dividen el comportamiento peatonal en tres fases: estratégica, táctica y operativa.

La fase estratégica se ocupa de la interacción con el medio ambiente, sin tener en cuenta a los otros agentes y su objetivo principal es guiar a los peatones a su objetivo. En la fase táctica, los agentes resuelven el problema de evasión de

colisiones e interactúan entre si. En la fase operativa, los agente determinan su dirección y la velocidad con la información de las etapas anteriores.

Para incrementar el nivel de realismo al realizar simulaciones sobre multitudes la comunicación entre los agentes involucrados es fundamental, ya que de esta manera, los comportamientos se vuelven más complejos y surgen comportamientos emergentes. El realismo de la simulación se ve beneficiado al incorporarse un modelo de comunicación ya que los agentes pueden cooperar para resolver alguna tarea específica o ayudarse entre todos para buscar algún tipo de recurso.

Pelechano y Badler proponen un modelo de comunicación y roles para los agentes [51]. Su sistema describe a los agentes, sus acciones y los resultados esperados al incorporar los roles de comunicación, además de proveer motivaciones, estrés, emociones, personalidad y decisiones. De esta forma es posible tener un sistema de alto nivel que se adapte a diversas situaciones, por ejemplo, permitir explorar ambientes desconocidos y encontrar una salida [26, 49].

2.5. Interacción de grupos

El hacer que la gente camine en grupos tiene dos objetivos principales, el primero y el más obvio es mejorar la calidad de la simulación. Observando a la gente en la calle, o en casi cualquier otro espacio abierto, la mayoría de las personas caminan en pequeños grupos. El segundo objetivo tiene implicaciones interesantes ya que es posible reducir el tiempo de procesamiento para el mismo número de agentes.

Durante el año 2010, Moussaid et al. analiza el movimiento de aproximadamente 1,500 grupos de peatones y señala que las multitudes son conformadas principalmente por grupos de personas que viajan juntas [52]. La mayoría de los trabajos de investigación en torno a la simulación de multitudes toma en cuenta sólo a individuos aislados. En situaciones reales hasta el 70 % de los peatones que constituyen un multitud se mueven en grupos, como amigos y familiares. Normalmente los grupos son de tamaño pequeño, con dos a cuatro miembros como los tamaños más frecuentes, mientras que los grupos más grandes son raros. Los grupos más grandes tienden a tener velocidades más bajas que los grupos de menor tamaño.

Los autores muestran que la densidad afecta a las formaciones en las que los grupos tienden a caminar. Cuando la densidad es baja en el área geográfica, miembros del grupo caminan uno a lado del otro; por otro lado, a medida que aumenta la densidad la formación lineal se dobla hacia delante convirtiéndose en un patrón semejante a una V , como se puede ver en la Figura 7. Estas formaciones están estrechamente relacionadas con las interacciones sociales, debido a que los miembros de un grupo tratan de mantener la comunicación entre si.

El siguiente modelo matemático, presentado por Moussaid et al. incluye fuerzas sociales para mantener unido al grupo:

$$\mathbf{f}_i^{grupo} = \mathbf{f}_i^{vis} + \mathbf{f}_i^{atr} + \mathbf{f}_i^{rep}$$

donde \mathbf{f}_i^{grupo} representa la respuesta del peatón i a otros miembros del grupo, \mathbf{f}_i^{vis} es la fuerza que mantiene a los miembros del grupo en el campo de visión,

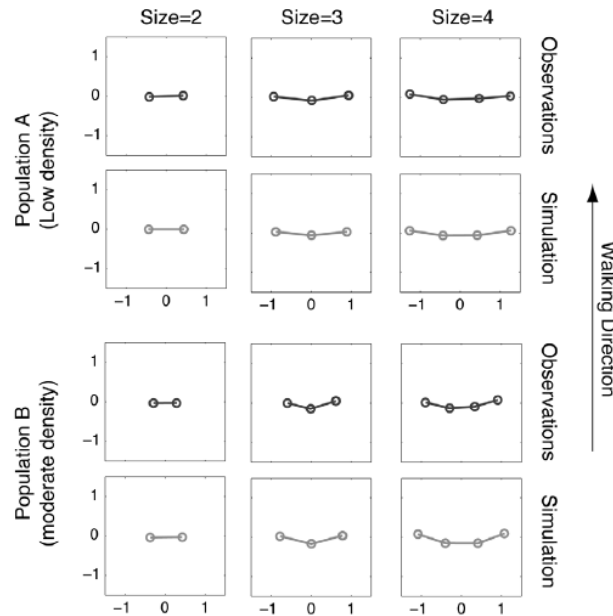


Figura 7. Patrones de organización de los grupos dependiendo de la densidad geográfica. Referencia original de [52].

f_i^{atr} es un término que dirige a los agentes hacia el centro de masa del grupo y f_i^{rep} es una fuerza de repulsión para mantener a los miembros del grupo separados.

Los individuos en una multitud se comportan de manera distinta cuando se encuentran solos o cuando forman parte de un grupo. Dentro de una multitud pueden ocurrir fenómenos como la polarización, la cual ocurre cuando dos o más grupos adoptan actitudes, opiniones o comportamientos divergentes. En ese caso es bastante probable que el comportamiento entre los grupos de la multitud resulte en confrontaciones si no se llega a un acuerdo.

En algunos escenarios es posible que la multitud busque oponentes. Un modelo dedicado a describir comportamientos grupales usando las relaciones intrínsecas de los elementos de dicho grupo es presentado en el trabajo de Thalmann [53]. Este modelo permite a los agentes reaccionar únicamente en la presencia de otros, por ejemplo cuando dos agentes se encuentran es posible evaluar los perfiles físicos y fisiológicos de cada uno y de los agentes a su alrededor; si resultan en un patrón similar, es probable que caminen juntos.

El anterior modelo incluye diferentes parámetros que son ajustables para los diversos grupos, además de ser posible establecer metas que cada grupo debe alcanzar. Otro de los parámetros que el modelo considera es el nivel de dominancia de cada grupo y finalmente el estado de ánimo. El estado de ánimo puede ser modificado si dos agentes se encuentran en un proceso aleatorio. Si un evento

como este ocurre, uno o varios agentes pueden ser seleccionados dependiendo del valor de dominancia y el resto de los agentes tendrá que disminuir su estado emocional. Thalmann concluye en que el comportamiento basado en las relaciones que existen dentro de los grupos de una multitud determinan el comportamiento general ésta, sin embargo, el modelo considera pocos parámetros de naturaleza sencilla, además de que se encuentra enfocado a multitudes genéricas.

La mayoría de los investigadores concuerdan en que es importante considerar la interacción de los grupos dentro de una multitud. Es por esta razón que se deben considerar protocolos de comunicación y sistemas que calculen colisiones, lo que puede ser un cuello de botella importante si no se aborda de forma adecuada. Muchos modelos se aseguran de optimizar esta etapa lo más posible [54].

Una estructura jerárquica para el manejo de grupos es presentado en 2010 [55]. Los autores proponen el uso de una base de datos de control, que es accesible por cada grupo y que se encarga del balanceo de estos cuando es necesario durante la simulación. Los autores utilizan un enfoque de líder-seguidor, existiendo un solo líder por grupo para el control de los agentes seguidor. Cuando se detecta una situación en el medio ambiente, las acciones se toman por grupo, no por individuo.

La teoría de la utilidad y la teoría de la comparación social han sido utilizadas para simular la dinámica de grupos [56]. Una desventaja de este enfoque es que no consideran que un grupo está formado por agentes que comparten un vínculo, como amigos o familiares, sino por agentes que comparten objetivos. La teoría de utilidad la emplean los agentes para unirse a un grupo con metas afines, mientras que la teoría de comparación social es utilizada para elegir a un miembro del grupo a seguir.

Park et al. presenta un sistema para simular miles de peatones en tiempo real con comportamientos sociales en 2012 [57]. Los grupos simulados en este trabajo están socialmente relacionados, es decir, constan de familiares y amigos. Los resultados muestran personajes en constante comunicación y exhiben muestras de comportamientos como esperar a otros miembros del grupo, grupos que se dividen y se reúnen y conversaciones entre los agentes.

3. Conclusión

El área de investigación de simulación de multitudes está en constante evolución y el número de disciplinas que cooperan con sus objetivos va en aumento. Este trabajo se ha centrado en el esfuerzo de la comunidad científica para generar los comportamientos de peatones con el fin de lograr simulaciones más realistas, pero también permitiendo una velocidad de fotogramas en tiempo real.

A pesar de que el uso del cómputo general en hardware gráfico (GPGPU) es relativamente nuevo, ya hay grupos de investigación que han diseñado nuevas técnicas para adherirse a la nueva tecnología. Muchas de estas técnicas permiten el desempeño eficiente de algoritmos que en arquitecturas secuenciales no lograrían tiempo real en las simulaciones.

Es evidente que los avances en la tecnología, específicamente en los procesadores gráficos, permiten a los investigadores escalar sus proyectos. El esfuerzo conjunto de estos proyectos tiene gran impacto en muchas áreas de la vida cotidiana.

Referencias

1. Wang, X., Zhang, J., Scalia, M.: Parallel Motion Simulation of Large-Scale Real-Time Crowd in a Hierarchical Environmental Model. *Mathematical Problems in Engineering* **2012** (2012) 1–15
2. Forster, M.: *World War Z* (2013)
3. Groenewegen, S.: Improving crowd behaviour for games and virtual worlds. In: *Proceedings of the Fifth International Conference on the Foundations of Digital Games - FDG '10*, New York, New York, USA, ACM Press (2010) 256–258
4. Wiß ner, M., Kistler, F., André, E.: Level of detail ai for virtual characters in games and simulation. *Motion in Games* (2010) 206–217
5. Szymanczyk, O., Dickinson, P., Duckett, T.: From Individual Characters to Large Crowds : Augmenting the Believability of Open-World Games through Exploring Social Emotion in Pedestrian Groups. In: *Think Design Play: DiGRA Conference*. (2011)
6. Wang, D., Pedreschi, D., Song, C., Giannotti, F., Barabasi, A.L.: Human mobility, social ties, and link prediction. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*, New York, New York, USA, ACM Press (2011) 1100
7. AFP: Chávez, ayatola y papa: los funerales que convocaron multitudes (2013)
8. Zhong, Z., Ding, N., Wu, X., Xu, Y.: Crowd surveillance using Markov Random Fields. In: *2008 IEEE International Conference on Automation and Logistics*. Number September, IEEE (September 2008) 1822–1828
9. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Number 2, IEEE (June 2009) 935–942
10. Cao, T., Wu, X., Guo, J., Yu, S., Xu, Y.: Abnormal crowd motion analysis. In: *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE (December 2009) 1709–1714
11. Husni, M., Suryana, N.: Crowd event detection in computer vision. In: *2010 2nd International Conference on Signal Processing Systems*, IEEE (July 2010) V1–444–V1–447
12. Zweng, A., Kampel, M.: Unexpected Human Behavior Recognition in Image Sequences Using Multiple Features. In: *2010 20th International Conference on Pattern Recognition*, IEEE (August 2010) 368–371
13. Xiong, G., Wu, X., Chen, Y.L., Ou, Y.: Abnormal crowd behavior detection based on the energy model. In: *2011 IEEE International Conference on Information and Automation*. Number June, IEEE (June 2011) 495–500
14. Li, N., Zhang, Z.: Abnormal Crowd Behavior Detection Using Topological Methods. In: *2011 12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, IEEE (July 2011) 13–18
15. Millan, E., Hernandez, B., Rudomin, I.: Large Crowds of Autonomous Animated Characters Using Fragment Shaders and Level of Detail. In Wolfgang Engel, ed.: *ShaderX5: Advanced Rendering Techniques*. Charles River Media (2006) 501—510

16. Thalmann, D., Grillon, H., Maim, J., Yersin, B.: Challenges in Crowd Simulation. In: 2009 International Conference on CyberWorlds, IEEE (2009) 1–12
17. Helbing, D., Farkas, I.J., Molnár, P., Vicsek, T.: Simulation of Pedestrian Crowds in Normal and Evacuation Situations. *Pedestrian and evacuation dynamics* **21** (2002)
18. Lemerancier, S., Jelic, A., Kulpa, R., Hua, J., Fehrenbach, J., Degond, P., Appert-Rolland, C., Donikian, S., Pettré, J.: Realistic following behaviors for crowd simulation. *Computer Graphics Forum* **31**(2pt2) (May 2012) 489–498
19. Sun, L., Qin, W.: Simulation of crowd behaviors based on event reaction. In: 2011 IEEE International Conference on Computer Science and Automation Engineering, IEEE (June 2011) 163–167
20. Luo, L., Zhou, S., Cai, W., Low, M.Y.H., Lees, M.: Toward a Generic Framework for Modeling Human Behaviors in Crowd Simulation. In: 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, IEEE (2009) 275–278
21. Zhang, L., Song, Y., Zhou, J., Gong, J., Li, W.: The framework of crowd simulation modeling with social network. In: 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). Number Iccasm, IEEE (October 2010) V3–437–V3–441
22. Kapadia, M., Singh, S., Reinman, G., Faloutsos, P.: A Behavior-Authoring Framework for Multiactor Simulations. *IEEE Computer Graphics and Applications* **31**(6) (November 2011) 45–55
23. Singh, S., Kapadia, M., Hewlett, B., Reinman, G., Faloutsos, P.: A modular framework for adaptive agent-based steering. In: Symposium on Interactive 3D Graphics and Games on - I3D '11. Volume 1., New York, New York, USA, ACM Press (2011) 141
24. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics* **21**(4) (August 1987) 25–34
25. Reynolds, C.W.: Steering behaviors for autonomous characters. In: 1999 Game Developers Conference. (1999) 763–782
26. Pelechano, N., Allbeck, J.M., Badler, N.I.: Controlling individual agents in high-density crowd simulation. In: Proceedings of the 2007 ACM SIGGRAPH. Volume 1., San Diego, California, Eurographics Association (2007) 99–108
27. Hadap, S., Eberle, D., Volino, P., Lin, M.C., Redon, S., Ericson, C.: Collision detection and proximity queries. In: Proceedings of the conference on SIGGRAPH 2004 course notes - GRAPH '04, New York, New York, USA, ACM Press (2004) 15–es
28. van den Berg, J., Manocha, D.: Reciprocal Velocity Obstacles for real-time multi-agent navigation. In: 2008 IEEE International Conference on Robotics and Automation, IEEE (May 2008) 1928–1935
29. Guy, S.J., Chhugani, J., Curtis, S., Dubey, P., Lin, M., Manocha, D.: PLEdestrians: a least-effort approach to crowd simulation. *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2010) 119–128
30. Choi, B., Komuravelli, R., Lu, V., Sung, H., Bocchino, R.L., Adve, S.V., Hart, J.C.: Parallel SAH k-D tree construction. *HPG '10 Proceedings of the Conference on High Performance Graphics* (2010) 77–86
31. dos Santos, A.L., Teixeira, J.M.X., de Farias, T.S., Teichrieb, V., Kelner, J.: kD-Tree Traversal Implementations for Ray Tracing on Massive Multiprocessors: A Comparative Study. In: 2009 21st International Symposium on Computer Architecture and High Performance Computing, IEEE (October 2009) 41–48

32. Bleiweiss, A.: Multi agent navigation on GPU. White paper, GDC (2009)
33. Jund, T., Kraemer, P., Cazier, D.: A unified structure for crowd simulation. *Computer Animation and Virtual Worlds* **23**(3-4) (May 2012) 311–320
34. Passos, E.B., Joselli, M., Zamith, M., Clua, E.W.G., Montenegro, A., Conci, A., Feijo, B.: A bidimensional data structure and spatial optimization for supermassive crowd simulation on GPU. *Computers in Entertainment* **7**(4) (December 2009) 1
35. Hoff, K.E., Zaferakis, A., Lin, M., Manocha, D.: Fast and simple 2D geometric proximity queries using graphics hardware. In: *Proceedings of the 2001 symposium on Interactive 3D graphics - SI3D '01*. Volume 31., New York, New York, USA, ACM Press (February 2001) 145–148
36. Sud, A., Govindaraju, N., Gayle, R., Andersen, E., Manocha, D.: Surface distance maps. In: *Proceedings of Graphics Interface 2007 on - GI '07*, New York, New York, USA, ACM Press (2007) 35
37. Sud, A., Andersen, E., Curtis, S., Lin, M., Manocha, D.: Real-time Path Planning for Virtual Agents in Dynamic Environments. In: *2007 IEEE Virtual Reality Conference, IEEE* (2007) 91–98
38. Sud, A., Andersen, E., Curtis, S., Lin, M.C., Manocha, D.: Real-time path planning in dynamic virtual environments using multiagent navigation graphs. *IEEE transactions on visualization and computer graphics* **14**(3) (2008) 526–38
39. Chenney, S.: Flow tiles. In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '04*, New York, New York, USA, ACM Press (2004) 233–242
40. Zhang, S., Li, M., Li, F., Liu, A., Cai, D.: A simulation model of pedestrian flow based on geographical cellular automata. In: *2011 19th International Conference on Geoinformatics*. Volume 2., IEEE (June 2011) 1–5
41. Klupfel, H.: The simulation of crowds at very large events. In: *Traffic and Granular Flow'05*. Springer Berlin Heidelberg (2007) 341–346
42. Treuille, A., Cooper, S., Popović, Z.: Continuum crowds. *ACM Transactions on Graphics* **25**(3) (July 2006) 1160
43. Fischer, L.G., Silveira, R., Nedel, L.: GPU Accelerated Path-Planning for Multi-agents in Virtual Environments. In: *2009 VIII Brazilian Symposium on Games and Digital Entertainment, IEEE* (2009) 101–110
44. Dapper, F., Prestes, E., Nedel, L.P.: Generating steering behaviors for virtual humanoids using BVP control. (2007)
45. Fiorini, P., Shiller, Z.: Motion Planning in Dynamic Environments Using Velocity Obstacles. *The International Journal of Robotics Research* **17**(7) (July 1998) 760–772
46. van den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. *Robotics Research* **70** (2011) 3–19
47. Wang, Y., Dubey, R., Magnenat-Thalmann, N., Thalmann, D.: An immersive multi-agent system for interactive applications. *The Visual Computer* **29**(5) (June 2012) 323–332
48. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. *Physical Review E* **51**(5) (May 1995) 4282–4286
49. Pelechano, N., Badler, N.I.: Modeling Crowd and Trained Leader Behavior during Building Evacuation. *IEEE Computer Graphics and Applications* **26**(6) (November 2006) 80–86
50. Beltaief, O., El Hadouaj, S., Ghedira, K.: Multi-agent simulation model of pedestrians crowd based on psychological theories. In: *2011 4th International Conference on Logistics, IEEE* (May 2011) 150–156

51. Pelechano, N., O'Brien, K., Silverman, B., Badler, N.I.: Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication. *First International Workshop on Crowd Simulation (2005)* **2**(1529-2401) (November 2005) 21–30
52. Moussaïd, M., Perozo, N., Garnier, S., Helbing, D., Theraulaz, G.: The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PloS one* **5**(4) (January 2010) e10047
53. Musse, S.R., Thalmann, D.: *Computer Animation and Simulation '97*. In Thalmann, D., Panne, M., eds.: *Computer Animation and Simulation '97*. Eurographics. Springer Vienna, Vienna (1997) 39–51
54. Musse, S.R., Ulicny, B., Aubel, A., Thalmann, D.: Groups and crowd simulation. In: *ACM SIGGRAPH 2005 Courses on - SIGGRAPH '05*, New York, New York, USA, ACM Press (2005) 8
55. He, C., Xiao, H., Dong, W., Deng, L.: Dynamic group behavior for real-time multi-agent crowd simulation. In: *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*. Volume 1., IEEE (February 2010) 544–546
56. Qiu, F., Hu, X.: Modeling Dynamic Groups for Agent-Based Pedestrian Crowd Simulations. In: *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE (August 2010) 461–464
57. Park, S.I., Peng, C., Quek, F., Cao, Y.: A Crowd Modeling Framework for Socially Plausible Animation Behaviors. In Kallmann, M., Bekris, K., eds.: *Motion in Games*. Volume 7660 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg (2012) 146–157

Hacia la formación automática de ontologías en documentos no estructurados

Martha Idalid Rivera González¹, Adolfo Guzmán Arenas² y Miguel González Mendoza¹

¹ Tecnológico de Monterrey Campus Estado de México,
Atizapán de Zaragoza, México
{A01371271, mgonza}@itesm.mx

² Centro de Investigación en Computación (CIC-IPN), Instituto Politécnico Nacional,
México DF, México
aguzman@ipn.mx

Resumen El presente trabajo pretende realizar una búsqueda, en una gran cantidad de documentos, iniciando con una palabra o concepto “semilla”, usando técnicas de minería de datos (minería de texto), se detectan palabras y frases temáticas, que son candidatos a ser conceptos (se usará desambiguación para pasarlos a conceptos aunque generalmente en campos especializados, cada palabra o frase temática tiene un significado único).

Palabras clave: ontologías, minería de datos, red semántica.

1. Introducción

En las últimas décadas han sido desarrollados numerosos enfoques, métodos y técnicas para el desarrollo en la búsqueda del nuevo conocimiento, las ontologías es una prueba de ello, que permite estructurar los datos donde la información es almacenada en nodos, agregando con ello la aplicación de métodos, permite realizar búsqueda de conocimientos de información no estructurada, tales como páginas web, documentos, artículos, palabras, entre otros.

La construcción de ontologías consiste en enriquecer o extender los nuevos conceptos obtenidos a partir de un corpus, utilizan modelos probabilísticos como el trabajo de Jian-hua Yeh [1] que utiliza LDA (Latent Dirichlet Allocation) para extraer los temas, con los cuales se construye aplicando el algoritmo de Clustering Aglomerativo Jerárquico [2].

La visualización de los datos es una fuente importante de ver los resultados. Se utiliza comúnmente en el análisis de datos para ayudar al usuario a obtener una primera idea acerca de la prima [10].

El experimento que se trabajó para el desarrollo del presente, es la recopilación de documentos relacionados con la palabra “fútbol” y los temas relacionados al mismo.

Este documento se organiza como sigue. La próxima Sección se dará una definición de ontología. La Sección 3 contiene la problemática. La Sección 4 se

presenta la metodología propuesta. La Sección 5 contiene los experimentos y resultados. Finalmente en la Sección 6 se presentan las conclusiones.

2. Ontologías

Una ontología define el vocabulario de un área mediante un conjunto de términos básicos y relaciones entre dichos términos, así como las reglas que combinan términos y relaciones que amplían las definiciones dadas en el vocabulario [3]. Una de las tareas importantes de las ontologías es el entendimiento del vocabulario. Otros autores definen una ontología como una especificación explícita de una conceptualización [4, 5], es decir proporciona una estructura y contenidos de forma explícita que codifica las reglas implícitas de una parte de la realidad, independientemente del fin y del dominio de la aplicación en el que se usarán o reutilizarán sus definiciones; Cuevas y Guzmán [6], como una estructura de datos donde la información es almacenada como nodos y relaciones, Zouaq y Nkambou [7] la define como la columna vertebral de la representación del conocimiento para la Web Semántica. Existen diversos trabajos relacionados con las ontologías que más adelante se abordará con más detalle.

Para el presente trabajo se utilizan diversas páginas Web seleccionadas mediante un término inicial o "semilla", los cuales se analizarán, buscando los conceptos comunes a dichos documentos (conceptos frecuentes) frases temáticas y las relaciones entre los conceptos.

3. Problemática

Dado muchos documentos de un mismo tema, encontrar de manera automática:

- a) Los nuevos conceptos que en ellos aparecen (conceptos relevantes a un documento), y
- b) las relaciones entre los conceptos.

Es decir, hallar de forma automática la ontología que describe los conceptos y las relaciones adyacentes en los documentos.

Como caso de estudio la semilla o tema buscado "fútbol", esto es, tenemos que hallar 100 nodos (conceptos) (luego 1,000) (luego diez mil) que se relacionen con "fútbol, balompié o soccer". El objetivo es encontrar nuevas relaciones con las palabras recientes de los documentos clave buscada (en algunos casos se encontrarán pares de términos sin relación). Esos nuevos conceptos hallados, más las relaciones entre ellos, forman una ontología (un modelo del conocimiento sobre "fútbol"), que es lo que finalmente se tiene que producir. La ontología que "rodea" a un concepto o semilla dado.

4. Metodología

Para el desarrollo del presente trabajo se realizará lo siguiente:

1.- Se recolectan muchos documentos (diez mil) documentos de un mismo tema (se usa una palabra semilla ejemplo futbol).

2.- Se hallan palabras (tokens) que ocurren frecuentemente en este documento:

a) Una palabra frecuente de orden "x"

Por cada palabra realizar la frecuencia en los documentos de orden x donde $x = 1..4$

Las palabras se contarán en forma conjunta dependiendo de su orden es decir "futbol" orden 1 , "futbol en" orden 2, "futbol en el" orden 3 , "futbol en el campo" orden 4

b) Se realiza la frecuencia

c) Se depura la información

d) Se realizan las frases temáticas hasta orden 4

3.- Se hallan las relaciones entre esos conceptos:

De ahí la necesidad de utilizar técnicas de clustering [8], [9] para ver la similitud de los conceptos.

4.- Se inicia con la recopilación de muchos documentos (8000) en formato txt como se muestra el diagrama siguiente posteriormente en un archivo se recopilan las palabras que no tienen ningún significado(a) posteriormente se buscan las palabras y se colocan las más frecuentes (b) posteriormente se separan las frases temáticas (c) y se forman otra lista con los conceptos y frases para su posterior análisis (d).

Se utiliza java como lenguaje de programación ya que facilita la creación de los hilos en los diversos procesos.

4.1. Algoritmo

a) Dado un conjunto finito de Documentos

$D_y \{, D_{y+1} D_n \dots \}$

$\square y = 1..n$ dónde $n=8000$ documentos

De tal forma que

$D_y \square T_{x+1,1}, T_{x+2,2} \dots T_{x+z,m}$

Dónde

x : fila del documento inicia de 1..m

m : tamaño de la longitud del token "y"

T_x : token (palabra) de cada uno de los documentos

z : token final de la fila "x"

b) Se divide los documentos en número de órdenes de tal forma que por cada documento con "y" tokens se enlistan en forma de columna de la sig. manera:

Orden 1

$T_{1,1}, T_{1,2}, T_{1,3} \dots T_{1,z}$

$T_{2,1} T_{2,2} T_{2,3} \dots T_{2,z}$

$T_{3,1} T_{3,2} T_{3,3} \dots T_{3,z}$

...

$$T_{x,1} T_{x,2} T_{x,3} \dots T_{x,z}$$

D_y	$\sum T_{x,z}$
$T_{1,1}$	
$T_{1,2}$	
$T_{1,3}$	
....	
$T_{1,z}$	
$T_{2,1}$	
$T_{2,2}$	
$T_{2,3}$	
...	
$T_{2,z}$	
...	
$T_{x,1} T_{x,2} T_{x,3} \dots T_{x,z}$	

$\sum T_{x,z}$ → Palabras

- c) Se realiza la frecuencia por cada token encontrado $T_{x,z}$
- d) Del total de los tokens encontrados se determina el porcentaje considerando un umbral del 0.01% para las palabras menos significativas y con ello se depura la información
- e) Se ordena los token de manera que se prepara para el orden siguiente. Ejemplo si es orden 2 se conjuntan dos tokens y se busca en todos los archivos la existencia repitiéndose inciso b y así sucesivamente hasta que se llegue al orden 4.
- d) Se presenta el algoritmo de las palabras encontradas orden “1”

```
// Algoritmo de encuentro de palabras orden 1
Thread hilo = new Thread(new Runnable() {
    @Override
    public void run() {
        String palabra = palabras.get(0).get(0).toString();
        String index = palabras.get(0).get(4).toString();
        Vector<Vector> ordenado = new Vector<Vector>();
        String palabra2 = null;
        String palabra1;
        for (int i = 0; i < datos.size(); i++) {
            Vector vector = datos.elementAt(i);
            int indexInicial = Integer.parseInt(vector.get(4).toString());
            if (indexInicial == 1) {
                for (StringTokenizer stringTokenizer =
                    new StringTokenizer(vector.get(0).toString());
                    stringTokenizer.hasMoreTokens(); ) {
                    palabra1 = stringTokenizer.nextToken();
                    palabra2 = stringTokenizer.nextToken();
                }
            }
        }
    }
});
```

Hacia la formación automática de ontologías en documentos no estructurados

```
ordenado.add(vector);
datos.removeElementAt(i);
break;
}
}
```

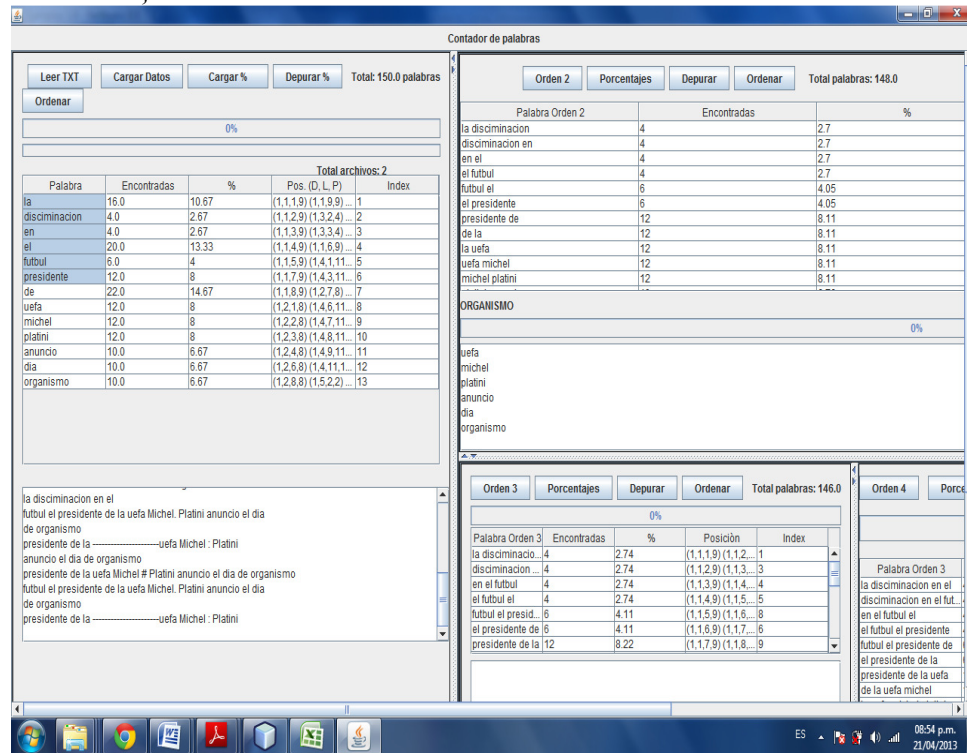


Figura 1. Vista general de la ejecución del sistema en Orden 1,2,3,4.

```
String primerPalabra = "";
String segundaPalabra = "";
Vector encontrado = new Vector();
Vector encontradoaux = new Vector();
int iaux = 0;
String auxSegundapalabra = null;
boolean ban = false;
for (int i = 0; i < datos.size(); i++) {
    encontrado = datos.elementAt(i);
    String palabras = encontrado.get(0).toString();
    int peso = Integer.parseInt(encontrado.get(4).toString());
    for (StringTokenizer stringTokenizer =
        new StringTokenizer(palabras);
```

```

stringTokenizer.hasMoreTokens();) {
primerPalabra = stringTokenizer.nextToken();
segundaPalabra = stringTokenizer.nextToken();
}
// System.out.println(primerPalabra + " " + segundaPalabra);

```

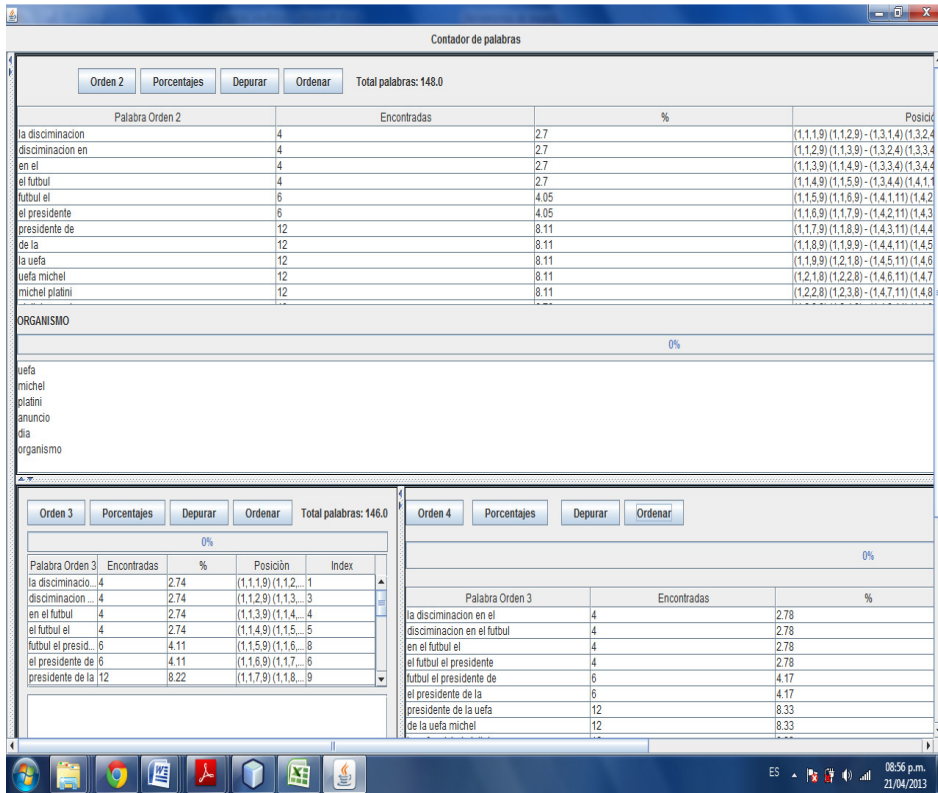


Figura 2. Pantalla del sistema.

5. Experimentación y resultados

A continuación se muestra una prueba de lo anterior

- Los documentos se depuran de tal forma que queden en formato txt.
- Se desarrolla parte de la programación dónde se concatenan los archivos participantes eligiéndolos en el botón Leer TXT como se muestran en la Fig. 1. Para el orden 1 (lado izquierdo)
- Se cargan los datos en el proceso determinándose las palabras encontradas
- se depuran la información con un umbral de 0.01% y se ordenan

e) Para el orden 2, 3 y 4 se realizan los pasos anteriores (Fig 2) nótese que al cambiar los ordenes las palabras se conjuntan en orden 2 dos palabras a buscar, si es para tres o cuatro sucesivamente

6. Conclusiones

El algoritmo para desarrollar la frecuencia en cada orden es precisa, ayuda a depurar las frecuencias que no son significativas, de esta manera al pasar al siguiente orden en la depuración de la información la frecuencia de las palabras encontradas se va haciendo cada vez menos, de esta forma al llegar al orden 4 la búsqueda será en menor tiempo. El lenguaje java permite trabajar muy bien la implementación de los hilos, con ello se gana tiempo en el proceso. Como trabajo futuro se pretende buscar la relación de los términos encontrados de “n” orden, entre los conceptos encontrados y con la frecuencia que se tiene buscar el seguimiento de los mismo, una de las soluciones propuestas es partir la oración y analizar cada uno de los conceptos.

Referencias

1. Jian-Hua Yeh and Naomi Yang. Ontology construction based on latent topic extraction in a digital library. In ICADL 08: Proceedings of the 11th International Conference on Asian Digital Libraries, pages 93–103, Berlin, Heidelberg, 2008. Springer-Verlag.
2. Dwi H. Widyantoro, Thomas R. Ioerger, and John Yen. An incremental approach to building a cluster hierarchy. In ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining, page 705, Washington, DC, USA, 2002. IEEE.
3. H. Phuc, L. Gauch, M. Speretta, Enriching Concept Description in an Amphibian Ontology with Vocabulary Extracted from wordnet, IEEE, Computer-Based Medical Systems, pp. 1-6, Aug. 2009.
4. Thomas R. Gruber, A translation approach to portable ontology specifications, Volume 5, Issue 2, Pag: 199 – 220, ISSN:1042-8143, 1993.
5. P. Buitelaar, P. Cimiano, B. Magnini, Ontology Learning from Text: Methods, Evaluation and Applications (Frontiers in Artificial Intelligence and Applications, Vol. 123), IOS Press, 2005.
6. Alma Delia Cuevas Rasgado and Adolfo Gúzman Arenas, A Language and Algorithm for Automatic Merging of Ontologies, Proceedings of the 15th International Conference on Computing, 2006.
7. Amal Zouaq, Roger Nkambou, "Evaluating the Generation of Domain Ontologies in the Knowledge Puzzle Project," IEEE Transactions on Knowledge and Data, Vol. 21 No. 11, pp. 1559-1572, Noviembre 2009.
8. Guojun Gan, Chaoqun Ma, Hunan Jianhong Wu, Data Clustering: Theory, Algorithms, and Applications, SIAM, Cambridge University Press, ISBN: 9780898716238, 2007
9. Sholom M. Weiss, Rutgers University and Nitin Indurkha, University of Sydney, softcover, ISBN 1-55860-403-0, 1997.
10. B. Fortuna, D. Mladenic, and M. Grobelnik. Visualization of text document corpus. Informatica Journal, 29(4):497–502.

Descripción de un proceso de enriquecimiento de ontologías obtenidas de base de datos relacionales

Joanna Alvarado-Uribe¹, David González-Marrón¹, Miguel González-Mendoza²
y Angélica Enciso-González¹

¹ Instituto Tecnológico de Pachuca, Pachuca de Soto, Hidalgo,
México

² Instituto Tecnológico y de Estudios Superiores de Monterrey,
Atizapán de Zaragoza, Edo. de México, México

joanna.1890@gmail.com, dgonzalez@itpachuca.edu.mx,
enciso_a@hotmail.com, mgonza@itesm.mx

Resumen En este trabajo se muestra la realización del proceso de enriquecimiento de la Ontología ISWC obtenida a partir de una Base de Datos Relacional; el cual permite transformar el Mundo Cerrado de las Bases de Datos Relacionales al Mundo Abierto de la Web Semántica por medio de Ontologías, mejorando la interoperabilidad de la información entre diferentes Bases de Datos con dominios comunes, las cuales no se encuentran relacionadas. La aplicación del proceso que en este trabajo se plantea, es una alternativa para lograr solucionar el problema de interoperabilidad, estableciendo una relación semántica entre dicha información. Se describe la realización del enriquecimiento utilizando el lenguaje D2RQ, el cual es utilizado para generar Ontologías por el método de mapeo directo de Bases de Datos Relacionales. Se presentan las ventajas de utilizar dos Ontologías propuestas denominadas ESOS y ESO, que facilitan y enriquecen el proceso de explotación de información semántica.

Palabras clave: Web semántica, ontologías, enriquecimiento.

1. Introducción

El proceso de enriquecimiento de Ontologías o también conocido como “Smushing”, es desarrollado con el principal propósito de permitir la interoperabilidad, entendiéndose por interoperabilidad, a la capacidad de dos o más herramientas, sistemas o componentes para trabajar conjuntamente en la realización de una tarea [3], o a la capacidad que tiene un producto o un sistema cuyas interfaces son totalmente conocidas para funcionar con otros productos o sistemas existentes o futuros, sin restricción de acceso o de implementación [1]. En este trabajo se concibe la interoperabilidad como la habilidad para integrar diferentes Bases de Datos que abordan dominios con características comunes, con el propósito de ampliar las áreas que individualmente abarcan. De igual manera, se busca transformar el enfoque de Mundo Cerrado de las Bases de Datos tradicionales a un Mundo Abierto con el uso de

Ontologías. Donde una Ontología define un vocabulario común que describe en un dominio específico, conceptos básicos y las relaciones entre ellos; convirtiéndose, en una pieza fundamental de las tecnologías orientadas a la Web Semántica. El proceso de enriquecimiento es realizado en este trabajo con la herramienta D2R Server. La cual permite la conversión de información relacional a semántica y la explotación de los datos semánticos utilizando SPARQL con un razonador denominado Snorql para hacer inferencia sobre los datos.

Para realizar este proceso, es necesario que las Bases de Datos Relacionales a las que se les aplicará el enriquecimiento, se encuentren normalizadas al menos en la Tercera Forma Normal (3FN), ya que si los datos se encuentran en un nivel de normalización menor, se encontrarán características indeseables que obstaculizarán el lograr la interoperabilidad, requiriéndose realizar un rediseño en los modelos de la Base de Datos, para cumplir con este requerimiento. A fin de facilitar el proceso de explotación de la información, se han construido dos Ontologías para ocultar la nomenclatura cerrada de las Bases de Datos Relacionales (una en español y otra en inglés), para mostrar al usuario una capa de presentación más comprensible, al generar sinónimos y agregaciones de atributos con términos menos técnicos, enfocados hacia usuarios no especializados en las Bases de Datos Relacionales, facilitando la creación de consultas para explotación de los datos.

A continuación, se describen los 7 apartados que conforman este artículo: 2) ¿Por qué enriquecer una Ontología obtenida de un mapeo automático de una Base de Datos Relacional (BDR)?, 3) Descripción del proceso de enriquecimiento, 4) Ontologías enriquecidas (ESO, ESOS), 5) Explotación de Ontologías enriquecidas, 6) Trabajos futuros y 7) Conclusiones.

2. ¿Por qué enriquecer una ontología obtenida de un mapeo automático de una base de datos relacional (BDR)?

Generalmente, las Bases de Datos Semánticas obtenidas del mapeo de Bases de Datos Relacionales, no tienen relación con otros dominios; en otras palabras, no se enlazan con Ontologías estandarizadas y reconocidas por la comunidad de la Web Semántica, como lo son: FOAF (The Friend Of A Friend) [2], VCARD (Electronic Card) [10], SKOS (Simple Knowledge Organization System) [26] y DC (Dublin Core) [24], por mencionar algunos ejemplos.

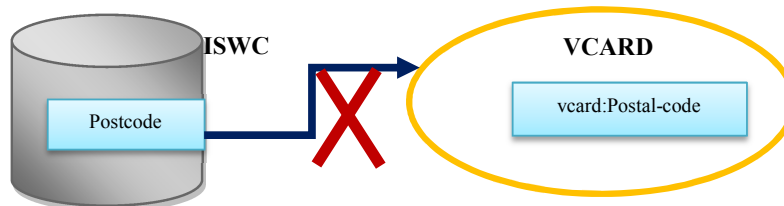


Figura 1. Bases de Datos Relacionales que no tienen relación con otros dominios.

Por lo que, a fin de lograr la interoperabilidad entre las Bases de Datos Semánticas, es requerido transformar el Mundo Cerrado de las BDR al Mundo Abierto de la Web Semántica, por medio del uso de Ontologías. De igual manera, el proceso de enriquecimiento permite generar o integrar Ontologías en las Bases de Datos Semánticas, con términos que describan claramente el significado de los mismos a los usuarios, facilitando la construcción de consultas. Lo anterior, se representa gráficamente en la Figura 1.

3. Descripción del proceso de enriquecimiento

El proceso de enriquecimiento de Ontologías se puede realizar de manera manual o automático, en este trabajo, se describe la realización de un proceso manual. Existen diferentes proyectos que se llevan a cabo para la realización de una aproximación automática, como el trabajo de Fink et al. [8], Shvaiko et al. [21], Simpson et al. [22], Martínez et al. [13], los cuales buscan la alineación de Ontologías de manera automática. En este capítulo, se presenta la descripción de los pasos que componen al proceso de enriquecimiento de Ontologías aplicado a información semántica obtenida de Base de Datos Relacionales; así como, un ejemplo del mismo.

3.1. Elección de la base de datos relacional

El proceso de enriquecimiento debe basarse en una Base de Datos Relacional, la cual pretende ser enriquecida con otros dominios o propiedades. Para fines demostrativos, se hace uso de la Base de Datos ISWC (siglas en inglés, International Semantic Web Conference), la cual administra principalmente: a) los datos de las conferencias o foros internacionales acerca de la Web Semántica [12]; b) las organizaciones involucradas en estas conferencias; c) los artículos acerca de la Web Semántica; y d) las personas interesadas en los temas que manejan los artículos.

3.2. Proceso de validación de la base de datos

Una vez seleccionada la Base de Datos Relacional, se verifica su nivel de normalización.

Se presenta el análisis realizado para verificar el nivel de normalización de la BDR ISWC y su procesamiento hasta lograr la Tercera Forma Normal, la cual es el nivel de normalización recomendado para la mayor parte de los modelos de Base de Datos. El modelo de la BD se representa utilizando el modelo Entidad-Relación de Chen [5], como puede ser visto en la Figura 2.

Entidades:

- **conferences** (ConfID, Name, URI, Date, Location, Datum)
- **organizations** (OrgID, Type, Name, Address, Location, Postcode, Country, URI, Belongsto, Homepage)
- **papers** (PaperID, Title, Abstract, URI, Year, Conference, Publish)

- **persons** (PerID, Type, FirstName, LastName, Address, Email, Homepage, Phone, URI, Photo)
 - **topics** (TopicID, TopicName, URI, ParentID)
- Relaciones:
- **rel_person_paper** (PersonID, PaperID)
 - **rel_paper_topic** (PaperID, TopicID, RelationType)
 - **rel_person_topic** (PersonID, TopicID)
 - **rel_person_organization** (PersonID, OrganizationID)

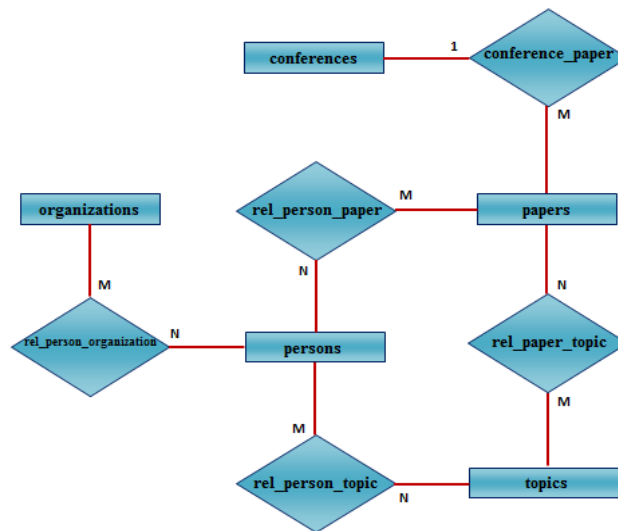


Figura 2. Diagrama Entidad – Relación de la Base de Datos ISWC.

3.3. Aplicación de las formas normales

3.3.1. Primera forma normal

La descripción del modelo que se presenta en la Figura 2, muestra que las tablas no tienen columnas repetidas, se identifica cada tabla con una clave única y las tablas fueron construidas de acuerdo con la similitud de las columnas; sin embargo, se realizan algunas modificaciones a los nombres de los campos con la finalidad de diferenciar las columnas que tienen el mismo nombre. Por ejemplo, Name se repite en las tablas “conferences” y “organizations”, por lo que para “conferences” Name se renombra a NameConf y para “organizations” Name se renombra a NameOrg.

También se debe considerar la atomicidad de los campos, el único campo que no se encuentra de manera atómica es “Address” de la tabla “organizations” ya que puede dividirse en Calle, Número Exterior, Número Interior, por citar algunos ejemplos; sin embargo, para fines de practicidad de este diseño y debido a que no se tiene la necesidad de indicar con claridad la dirección, ya que la Base de Datos indica que la comunicación que se tiene con las organizaciones es más de manera vía electrónica en

lugar del correo tradicional, el campo no se atomiza; además, la información contenida en este campo es de poca utilidad al momento de interrogar la Base de Datos.

Las modificaciones que se efectúan en los nombres de los campos se marcan en color gris.

Tablas:

- **conferences** (ConfID, NameConf, URISConf, Date, LocationConf, Datum)
- **organizations** (OrgID, TypeOrg, NameOrg, AddressOrg, LocationOrg, Postcode, Country, URIOrg, Belongsto, HomepageOrg)
- **papers** (PaperID, Title, Abstract, URIPaper, Year, Conference, Publish)
- **persons** (PerID, TypePer, FirstName, LastName, AddressPer, Email, HomepagePer, Phone, URIPer, Photo)
- **topics** (TopicID, TopicName, URITopic, ParentID)

3.3.2. Segunda forma normal

En esta Forma se verifica que los registros no tengan valores repetidos y que los campos sean dependientes de su clave primaria; si se presenta el caso, se realiza una nueva tabla con estos valores y campos, y las tablas se enlazan por medio de una clave foránea o externa. Por lo que, los campos “Postcode” y “Country” son eliminados de la tabla “organizations” para formar una nueva tabla denominada “Countries”.

Tabla:

- **organizations** (OrgID, TypeOrg, NameOrg, AddressOrg, LocationOrg, Postcode, Country, URIOrg, Belongsto, HomepageOrg)

Se agrega la tabla “countries” y se modifica la tabla “organizations” al agregar el campo “CountryID” para enlazar ambas tablas con la llave primaria de la tabla “countries”:

- **countries** (CountryID, CountryName, Postcode)
- **organizations** (OrgID, TypeOrg, NameOrg, AddressOrg, LocationOrg, URIOrg, Belongsto, HomepageOrg, CountryID)

Por lo que las tablas quedan estructuradas de la siguiente manera:

- **conferences** (ConfID, NameConf, URISConf, Date, LocationConf, Datum)
- **organizations** (OrgID, TypeOrg, NameOrg, AddressOrg, LocationOrg, URIOrg, Belongsto, HomepageOrg, CountryID)
- **papers** (PaperID, Title, Abstract, URIPaper, Year, Conference, Publish)
- **persons** (PerID, TypePer, FirstName, LastName, AddressPer, Email, HomepagePer, Phone, URIPer, Photo)
- **topics** (TopicID, TopicName, URITopic, ParentID)
- **countries** (CountryID, CountryName, Postcode)
- **rel_person_paper** (PersonID, PaperID)
- **rel_paper_topic** (PaperID, TopicID, RelationType)
- **rel_person_topic** (PersonID, TopicID)

- **rel_person_organization** (PersonID, OrganizationID)

En la Tabla 1, se describe la relación que se tiene de los campos entre su llave primaria, lo que demuestra que cada campo es dependiente de su llave primaria.

Tabla 1. Relación de los campos con su llave primaria.

<p>Tabla conferences</p> <ul style="list-style-type: none"> • conferences.ConfID ---> NameConf • conferences.ConfID ---> URICConf • conferences.ConfID ---> Date • conferences.ConfID ---> LocationConf • conferences.ConfID ---> Datum 	<p>Tabla organizations</p> <ul style="list-style-type: none"> • organizations.OrgID ---> TypeOrg • organizations.OrgID ---> NameOrg • organizations.OrgID ---> AddressOrg • organizations.OrgID ---> LocationOrg • organizations.OrgID ---> URIOrg • organizations.OrgID ---> Belongsto • organizations.OrgID ---> HomepageOrg
<p>Tabla papers</p> <ul style="list-style-type: none"> • papers.PaperID ---> Title • papers.PaperID ---> Abstract • papers.PaperID ---> URIPaper • papers.PaperID ---> Year • papers.PaperID ---> Conference • papers.PaperID ---> Publish 	<p>Tabla persons</p> <ul style="list-style-type: none"> • persons.PerID ---> TypePer • persons.PerID ---> FirstName • persons.PerID ---> LastName • persons.PerID ---> AddressPer • persons.PerID ---> Email • persons.PerID ---> HomepagePer • persons.PerID ---> Phone • persons.PerID ---> URIPer • persons.PerID ---> Photo
<p>Tabla topics</p> <ul style="list-style-type: none"> • topics.TopicID ---> TopicName • topics.TopicID --->URITopic • topics.TopicID ---> ParentID 	<p>Tabla countries</p> <ul style="list-style-type: none"> • countries.CountryID ---> CountryName • countries.CountryID ---> Postcode

3.3.3. Tercera forma normal

Una vez que se construyeron las tablas y se establecieron las relaciones entre ellas, se verifica que los campos dependan de la llave; en el caso contrario, se debe crear una nueva tabla con estos valores.

Las tablas normalizadas hasta la 2FN son:

- **conferences** (ConfID, NameConf, URICConf, Date, LocationConf, Datum)
- **organizations** (OrgID, TypeOrg, NameOrg, AddressOrg, LocationOrg, URIOrg, Belongsto, HomepageOrg, CountryID)
- **papers** (PaperID, Title, Abstract, URIPaper, Year, Conference, Publish)
- **persons** (PerID, TypePer, FirstName, LastName, AddressPer, Email, HomepagePer, Phone, URIPer, Photo)
- **topics** (TopicID, TopicName, URITopic, ParentID)
- **countries** (CountryID, CountryName, Postcode)

Por lo que, para la 3FN se obtiene el esquema de la Figura 3.

Analizando las tablas, todos los campos son dependientes de la llave; por lo que, no es necesario dividirlos.

Se diseña nuevamente el esquema de la Base de Datos Relacional, con los cambios realizados en cada Forma Normal, tal como se muestra en la Figura 4.

Descripción de un proceso de enriquecimiento de ontologías obtenidas...

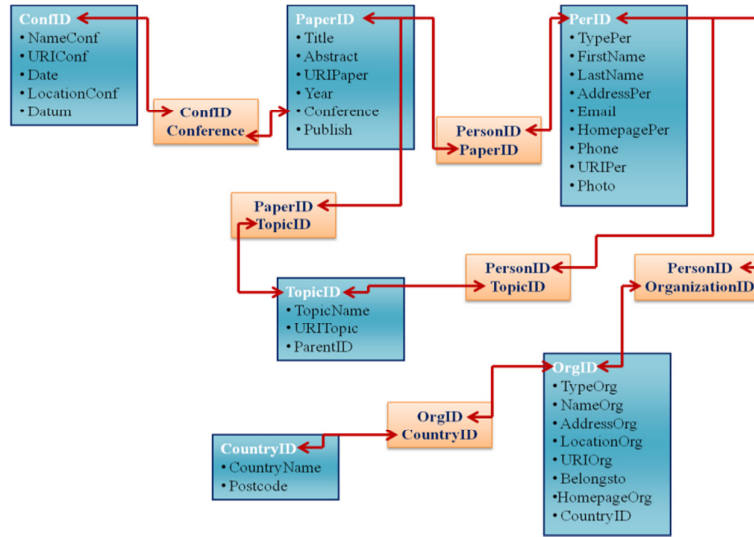


Figura 3. Representación de las llaves primarias y foráneas para la 3FN.

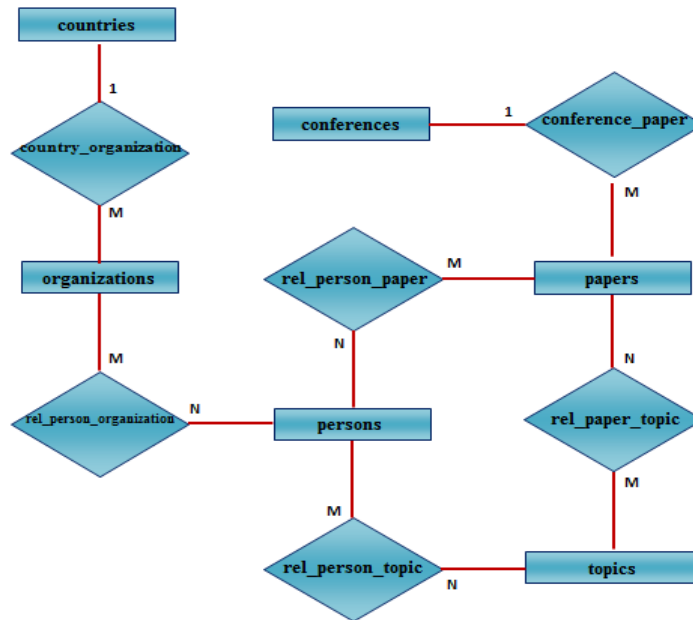


Figura 4. Diagrama Entidad – Relación de la Base de Datos ISWC Normalizado.

Con el diagrama anterior, se ilustra que se ha alcanzado el máximo nivel de normalización de la Base de Datos Relacional ISWC, por lo que se puede proceder a realizar el paso 3.

3.4. Herramienta de mapeo

Una vez que se normaliza la Base de Datos Relacional, por medio de la herramienta D2R Server, se realiza el mapeo a la Base de Datos. Para ello, se realizan las siguientes instrucciones:

- a) Se ejecuta el Símbolo del Sistema.
- b) Se tecléa CD y se copia la dirección que indica la ubicación del programa D2R Server, se da Enter.
- c) Se tecléa la instrucción [4]:
`generate-mapping -o NombreArchivoMapeo.ttl -u usuario -p password
jdbc:mysql://localhost/NombreBaseDeDatosRelacional`
- d) Una vez tecléada la instrucción, se da Enter.

Dentro de la carpeta del programa D2R Server se crea un archivo de salida, con el nombre y la extensión que se indicó en la instrucción; este archivo es el que se utilizará para hacer el enriquecimiento de la ahora Ontología.

3.5. Análisis de los campos de la base de datos relacional

Se realiza un análisis al diccionario de datos, en el caso de que se tenga, de la Base de Datos Relacional elegida. O de lo contrario, se analizan los campos que la conforman y los registros de los mismos. Por razones de espacio, se omite la descripción de los campos.

3.6. Estudio de ontologías reconocidas por la comunidad de la web semántica

Se realiza un estudio de las Ontologías estandarizadas existentes para conocer su dominio y las propiedades que las conforman.

De acuerdo con la investigación realizada, se obtuvieron que las Ontologías más usadas y reconocidas por la comunidad de la Web Semántica son:

- FOAF (The Friend Of A Friend).
Esta Ontología se utiliza para describir a las personas y sus relaciones sociales en la Web. Se centra principalmente en la existencia de las personas en el mundo virtual, con muchas propiedades relacionadas con la actividad en línea o identidad. Por ejemplo, foaf:mbox, foaf:SkypeID, foaf:msnID, foaf:geekcode, etc. Nada acerca de las relaciones familiares, la dirección física, entre otros aspectos. Proporciona información sobre organizaciones o grupos con un enfoque similar sobre su existencia en la Web (página Web del lugar de trabajo). Es especialmente adecuado para describir a las personas en la Web basados en plataformas sociales (facebook, twitter, blogspot...).
- VCARD (Electronic Card).

Ontología comúnmente utilizada para describir a las personas.

- DC (Dublin Core).
Ontología relacionada con la creación de un recurso; por ejemplo, un artículo, libro u otras publicaciones. Y los datos relacionados a su divulgación.
- ISWC (International Semantic Web Conference).
Ontología creada para el manejo de términos relacionados con el Foro de ISWC.
- SKOS (Simple Knowledge Organization System).
Es un modelo para expresar la estructura básica de "esquemas conceptuales" mediante jerarquías simples amplios (broader) / estrechos (narrower) (sin semántica formal). También se utiliza para representar listas de términos, vocabularios controlados y tesauros.
- OWL (Ontology Web Language).
Ontología que establece reglas para la edición de otras Ontologías.

3.7. Estudio de las propiedades de las ontologías

Se estudian las descripciones de las propiedades que conforman la Ontología, en el caso de que se tengan, o el tipo de dato que representa cada propiedad.

Se muestran dos ejemplos, en la Tabla 2, de la Ontología DC.

Tabla 2. Propiedades de la Ontología DC.

dc:creator La entidad encargada principalmente de la creación del recurso.		dc:language El lenguaje del recurso.	
Type:	--	Type:	--
Domain:	rdfs:Resource,	Domain:	rdfs:Resource,
Range:	xsd:string,	Range:	xsd:string,
Superproperties:	--	Superproperties:	--
Subproperties:	--	Subproperties:	--
Cardinality:	1	Cardinality:	1

3.8. Establecer la relación de la ontología obtenida de la base de datos relacional y las ontologías

Se identifican las relaciones entre las propiedades de la Ontología no estandarizada con las propiedades de las Ontologías estandarizadas. Se puede realizar una tabla en forma de concentrado para registrar las respectivas correspondencias, como se ilustra en la Tabla 3.

Tabla 3. Relación entre los campos de la BDR ISWC y las propiedades de las Ontologías seleccionadas.

Columna	Descripción	Ontología (Propiedad)
LocationOrg	Localidad de la Organización	v:Location
Postcode	Código Postal del País	v:postal-code

Columna	Descripción	Ontología (Propiedad)
CountryName	Nombre del País	v:country-name
Year	Año del Artículo	dc:date
FirstName	Nombre de la Persona	foaf:name
LastName	Apellido Paterno de la Persona	
AddressPer	Dirección de la persona	v:Address
Email	Correo Electrónico de la Persona	foaf: mbox

3.9. Evaluar la similitud entre las propiedades

Se realiza una evaluación de los términos y los contenidos, para determinar la similitud entre las propiedades de acuerdo con el esquema de la Figura 5:

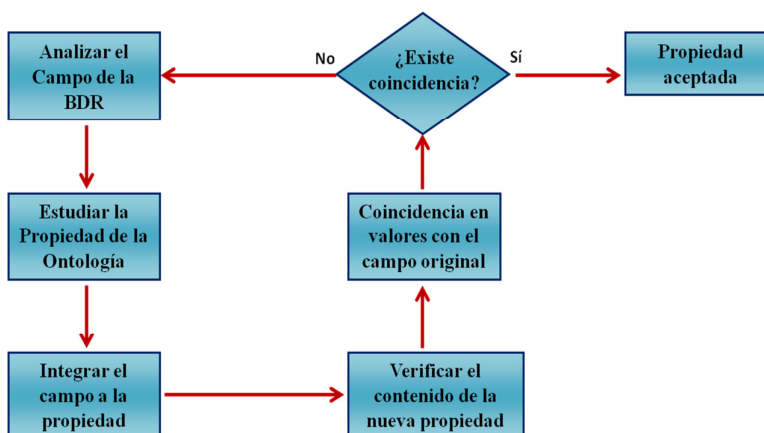


Figura 5. Pasos para realizar la evaluación de similitud entre las propiedades.

Se muestran dos ejemplos con propiedades existentes y la propuesta de la propiedad nueva. Para la propiedad existente `FirstName` de la tabla “persons” la propuesta del nombre de la nueva propiedad es: `foaf:name`; ya que por medio de esta propiedad se pueden unir el campo `FirstName` y `LastName` brindando un solo resultado al usuario al consultar esta propiedad; lo que le facilita la extracción de datos. Y para la propiedad existente `Address` de la tabla “organizations”, la propuesta del nombre de la nueva propiedad es: `v:Address`; ya que los nombres son idénticos para ambas propiedades, lo único que cambia es la Ontología a la que pertenecen.

3.10. Integrar las propiedades de la(s) ontología(s) a la ontología obtenida

Se integran las propiedades a la Ontología no estandarizada. Esta integración se realiza en el archivo de mapeo con extensión `.ttl` de manera manual, siguiendo estos pasos:

- 1) Abrir el archivo “ArchivoDeMapeo.ttl” con el programa Adobe Dreamweaver CS4, por ejemplo.

- 2) Agregar en la parte de @PREFIX la siguiente línea: **@prefix esos: <esos/> .**
- 3) Ya que por medio de esta línea podemos agregar una nueva Ontología a nuestro archivo de mapeo; la Ontología que se va a integrar es la de ESOS.
- 4) Para agregar las propiedades, se debe estudiar cómo se declaran las propiedades; mediante las Figuras 6 y 7 se explica esta actividad.

```
# Table conferences
map:Conferences a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "conferences/@@conferences.ConfID@";
  d2rq:class iswc:Conference;

# Some unrelated properties to test class definitions
d2rq:classDefinitionLabel "conference";
d2rq:classDefinitionComment "A conference";
d2rq:additionalClassDefinitionProperty map:conferenceSubClassOf;
.

map:conferences_Name a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:Conferences;
  d2rq:property rdfs:label;
  d2rq:column "conferences.Name";

# Some properties to test property definitions
d2rq:propertyDefinitionLabel "label";
d2rq:propertyDefinitionComment "A human-readable name for the subject.";
d2rq:additionalPropertyDefinitionProperty map:labelDomain;
.
```

Figura 6. Parte del archivo de mapeo .ttl de la Base de Datos ISWC.

El recuadro rojo, nos indica las líneas que se utilizan para agregar una nueva propiedad:

```
map:conferences_Name a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:Conferences;
  d2rq:property rdfs:label;
  d2rq:column "conferences.Name";
```

Donde:

- map:conferences_Name a d2rq:PropertyBridge;
conferences_Name es el nombre de la tabla y la columna; *map* y *a* *d2rq:PropertyBridge* son palabras reservadas.
- d2rq:belongsToClassMap map:Conferences;
d2rq:belongsToClassMap *map:* son palabras reservadas y *Conferences* es la tabla a la que se hace referencia.
- d2rq:property rdfs:label;
d2rq:property son palabras reservadas y *rdfs:label* es el nombre de la propiedad que se va a cambiar de acuerdo con la propiedad que se desea agregar.
- d2rq:column "conferences.Name";
d2rq:column son palabras reservadas y "*conferences.Name*" es la parte conformada por "Tabla.Columna", no se deben olvidar las comillas.
- . (*punto*)

Indica que han terminado las sentencias declaratorias de una propiedad.

```
# Ontologia ESOS
# Table conferences
map:conferences_Name a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:Conferences;
  d2rq:property esos:nombre_c;
  d2rq:column "conferences.Name";
.
map:conferences_Fecha a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:Conferences;
  d2rq:property esos:fecha_c;
  d2rq:column "conferences.Date";
.
map:conferences_Ubicacion a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:Conferences;
  d2rq:property esos:ubicacion_c;
  d2rq:column "conferences.Location";
.
```

Figura 7. Ejemplo de la declaración de nuevas propiedades.

Ya que se identificaron estas partes, se procede a integrar las nuevas propiedades, tal como se muestra en la Figura 7.

Una vez que se agregaron todas las propiedades, se Guardan los cambios.

3.11. Verificar la integración de las propiedades

Se realizan consultas en Snorql para verificar que la integración de las propiedades se haya realizado con éxito, comparando los valores de las nuevas propiedades con los de las existentes.

Para estas pruebas, se recomienda, realizar las comparaciones propiedad por propiedad (propiedad original vs propiedad nueva), con la finalidad de corroborar que todas las propiedades fueron integradas correctamente; esta actividad se muestra con mayor detalle en el punto 5.

3.12. Depurar la integración de las propiedades

En el caso de que las pruebas a las nuevas propiedades arrojen resultados diferentes de los valores que tienen las propiedades existentes, se modifican las propiedades que no muestren los resultados esperados; repitiendo el proceso desde el paso 3.4.

En este caso, no se tuvieron propiedades adaptadas de manera errónea, todas las propiedades brindaron los resultados esperados; por lo que, se concluye que fue una integración correcta.

4. Ontologías enriquecidas (ESO, ESOS)

Se crearon dos nuevas Ontologías, denominadas ESOS (siglas en inglés de Exploitation Semantic Ontology in Spanish, se traduce como Ontología de Explotación Semántica en Español); Ontología realizada para indicar las propiedades que conforman a la Ontología ISWC en el idioma español. Y ESO (siglas en inglés de Exploitation Semantic Ontology, se traduce como Ontología de Explotación Semántica), con la finalidad de facilitar al usuario, que no tiene conocimientos sobre Ontologías, la construcción de consultas; aunque una desventaja es el decremento de la interoperabilidad entre Bases de Datos.

En la Tabla 4, se dan ejemplos de las propiedades originales de la Ontología y el nombre que se adopta en estas nuevas Ontologías.

Tabla 4. Representación de las Ontologías de Explotación ESOS y ESO para ISWC.

ID	Propiedad Original	Nuevo Nombre	Descripción	Ontología
1	rdfs:label	esos:nombre_c	Nombre de la Conferencia	ESOS
2	dc:date	esos:fecha_c	Fecha de la Conferencia	ESOS
3	iswc:location	esos:ubicacion_c	Ubicación de la Conferencia	ESOS
4	rdf:type	esos:tipo_org	Tipo de la Organización	ESOS
5	rdfs:label	esos:nombre_org	Nombre de la Organización	ESOS
6	rdfs:label	eso:name_c	Nombre de la Conferencia	ESO
7	dc:date	eso:date_c	Fecha de la Conferencia	ESO
8	iswc:location	eso:location_c	Ubicación de la Conferencia	ESO
9	rdf:type	eso:type_org	Tipo de la Organización	ESO
10	rdfs:label	eso:name_org	Nombre de la Organización	ESO

5. Explotación de ontologías enriquecidas

Para verificar que la información obtenida anteriormente sea correcta, se realiza el ejemplo de una misma consulta, utilizando:

- 1) El archivo de mapeo de manera primitiva, el cual se distingue por el prefijo vocab precedido de cada propiedad.
- 2) El archivo de mapeo enriquecido con las Ontologías FOAF, VCARD, DC, SKOS, por mencionar algunas de ellas; dicho esquema se denomina como Ontología ISWC.
- 3) El archivo de mapeo enriquecido con las Ontologías anteriores y dos nuevas Ontologías nombradas: ESOS y ESO.

La consulta que se ejemplificará en las 3 formas descritas anteriormente es: Seleccionar todas las organizaciones donde la gente interesada en “Semantic Web” trabaje. Utilizando la función COUNT para conocer el número de organizaciones [7].

Primera consulta: El archivo de mapeo que se utiliza para ejemplificar esta consulta es el obtenido directamente de mapear la Base de Datos Relacional ISWC; el cual se distingue al visualizar en las propiedades al prefijo *vocab*; vocab representa una

Ontología primitiva que se obtiene de manera predeterminada al realizar el proceso de mapeo a la BDR. En la ventana del razonador Snorql, utilizando el archivo de mapeo primitivo, se visualiza la consulta y los resultados como se muestra en la Figura 8.

The screenshot shows the Snorql interface with a SPARQL query and its results. The query is as follows:

```
SPARQL:
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX db: <http://localhost:2020/resource/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX map: <file://C:/davidn/Experimentos/d2r-server-0.7/iswc_j.map#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vocab: <http://localhost:2020/resource/vocab/>

SELECT COUNT(?nombreOrganizacion) WHERE {
  ?idPersona vocab:persons_FirstName ?nombrePersona.
  ?idPersona vocab:persons_LastName ?apellidoPersona.
  ?idTema vocab:rel_person_topic ?idPersona.
  ?idTema vocab:topics_TopicName ?nombreTema .
  FILTER (?nombreTema = "Semantic Web").
  ?idPersona vocab:rel_person_organization ?organization .
  ?organization vocab:organizations_NameOrg ?nombreOrganizacion
}
```

Results:

SPARQL results:

.1
7

Figura 8. Consulta ejecutada en Snorql usando la Ontología vocab.

The screenshot shows the Snorql interface with a SPARQL query and its results. The query is as follows:

```
SPARQL:
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX iswc: <http://annotation.semanticweb.org/iswc/iswc.daml#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX eso: <http://localhost:2020/resource/eso/>
PREFIX map: <file://Users/richard/D2RQ/workspace/D2RQ/doc/example/mapping-iswc.n3#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vocab: <http://localhost:2020/resource/vocab/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX esos: <http://localhost:2020/resource/esos/>

SELECT COUNT(?nombreOrganizacion) WHERE {
  ?idPersona foaf:name ?nombrePersona.
  ?idPersona iswc:research_interests ?idTema.
  ?idTema rdfs:label ?nombreTema .
  FILTER (?nombreTema = "Semantic Web").
  ?idPersona iswc:has_affiliation ?organization .
  ?organization rdfs:label ?nombreOrganizacion
}
```

Results:

SPARQL results:

.1
7

Figura 9. Consulta ejecutada en Snorql usando la Ontología ISWC.

La construcción de esta consulta por medio de la Ontología vocab no es muy amigable para el usuario, ya que tendría que conocer la estructura de la Base de Datos para poder construirla. Por lo anterior, se realiza el enriquecimiento del archivo de mapeo con las Ontologías estandarizadas y reconocidas por la comunidad semántica, como se muestra en el inciso b).

Segunda consulta: Se realiza el enriquecimiento de la Ontología ISWC con las Ontologías estandarizadas por la comunidad semántica, para permitir la interoperabilidad entre Bases de Datos y facilitar la explotación de la información. En la ventana del razonador Snorql, utilizando la Ontología ISWC, se visualiza la consulta y los resultados como se muestra en la Figura 9.

Con este enriquecimiento, alcanzamos el objetivo de lograr la interoperabilidad entre Bases de Datos al hacer uso de Ontologías reconocidas por la comunidad de la Web Semántica y a su vez, se facilita al usuario la construcción de consultas; aunque, si no se conocen los términos que cada Ontología maneja, el usuario también podría presentar duda al momento de indicar las propiedades necesarias en la consulta requerida. Por lo que, el siguiente paso es construir dos Ontologías que le faciliten al usuario el desarrollo de consultas.

```

SPARQL:
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX iswc: <http://annotation.semanticweb.org/iswc/iswc.daml#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX eso: <http://localhost:2020/resource/eso/>
PREFIX map: <file://Users/richard/D2RQ/workspace/D2RQ/doc/example/mapping-iswc.n3#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vocab: <http://localhost:2020/resource/vocab/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX esos: <http://localhost:2020/resource/esos/>

SELECT count(?nombreOrganizacion) WHERE {
  ?idPersona esos:nombre_pers ?nombrePersona.
  ?idPersona iswc:research_interests ?idTema.
  ?idTema esos:nombre_tema ?nombreTema .
  FILTER (?nombreTema = "Semantic Web").
  ?idPersona iswc:has_affiliation ?organizacion .
  ?organizacion esos:nombre_org ?nombreOrganizacion
}

Results:   

SPARQL results:
.1
7
    
```

Figura 10. Consulta ejecutada en Snorql usando la Ontología ESOS obtenida de la Base de Datos ISWC.

Tercera consulta: Se realiza la creación de dos Ontologías denominadas ESOS y ESO, con la finalidad de facilitar al usuario, que no tiene conocimientos sobre Ontologías, la construcción de consultas. Mediante las Ontologías ESOS y ESO, los

resultados son los mismos, lo único que cambian son las propiedades que enlazan los sujetos con los objetos; las cuales hacen comprensible la obtención de los datos de la consulta, ver Figura 10.

La utilización de las nuevas Ontologías, es recomendada para usuarios con poca experiencia en las Ontologías estandarizadas.

Por medio del ejercicio, se visualiza que la información fue generada correctamente de la Base de Datos Relacional a Semántica, logrando con ello, incorporar nuevas Ontologías a su dominio y a su vez, permitir la interoperabilidad al cambiar del Mundo Cerrado a un enfoque abierto.

Para concluir con este capítulo, se realiza un cuadro comparativo de los tres ejercicios propuestos, enlistando sus ventajas y desventajas, ver Tabla 5.

Tabla 5. Cuadro comparativo de las consultas propuestas.

Cuadro comparativo		
Primera Consulta	Segunda Consulta	Tercera Consulta
<p><u>Ventajas</u></p> <p>3 Los creadores de la Base de Datos Relacional, tienen mayor facilidad para relacionar los términos que se han mapeado y así poder explotar la información de manera semántica.</p> <p><u>Desventajas</u></p> <p>4 Para un usuario no especializado en el dominio de la Base de Datos Relacional, la construcción de una consulta para explotar la información, consumirá más tiempo, comparado con el usuario experto, ya que los términos no le son amigables.</p> <p>5 No se tiene interoperabilidad entre Bases de Datos Relacionales.</p> <p>6 Se mantiene el Mundo Cerrado.</p>	<p><u>Ventajas</u></p> <p>7 Los usuarios conocedores de las Ontologías estandarizadas que en este trabajo se presentan, podrán fácilmente localizar los términos necesarios para la construcción de una consulta.</p> <p>8 Al enriquecer el archivo con Ontologías estandarizadas, se permite la interoperabilidad entre las Bases de Datos Relacionales.</p> <p>9 Se cambia el enfoque al Mundo Abierto.</p> <p><u>Desventajas</u></p> <p>10 Para un usuario que desconozca las Ontologías estandarizadas, se tendrá dificultades al momento de construir una consulta.</p>	<p><u>Ventajas</u></p> <p>11 Los usuarios no especializados en el dominio de la Base de Datos Relacional ni en las Ontologías estandarizadas, realizan fácilmente la explotación de la información.</p> <p>12 Se mantiene el enfoque al Mundo Abierto.</p> <p><u>Desventajas</u></p> <p>13 Existen más términos equivalentes para la realización de las consultas, lo cual podría confundir más al usuario al momento de formular sus consultas.</p>

6. Trabajos futuros

- 1) Se propone la realización de una aplicación que permita crear nuevas Ontologías con base en una Base de Datos Relacional; que asocie Ontologías estandarizadas y establezca sinónimos para términos especializados, en un lenguaje común.
- 2) Desarrollar herramientas gráficas para archivos mapeados en D2RQ, ya que los actuales, como pueden ser: Swoop o Protégé; están más orientados a trabajar con Ontologías mapeadas en los lenguajes OWL y RDF.

- 3) Realizar aplicaciones que permitan la construcción de consultas con mayor facilidad, ya que SPARQL requiere actualmente, un análisis profundo del conocimiento de la organización de los datos para generar las consultas.

7. Conclusiones

La Web Semántica (WS) es un área muy amplia, que necesita de una profunda investigación para su implementación, muchas áreas deben conjuntarse para su adaptación a la generalidad de usuarios existentes.

El proceso de enriquecimiento descrito, permite mejorar la interoperabilidad y la explotación de información semántica obtenida de fuentes de datos relacionales; la realización automática del enriquecimiento es todavía un problema abierto en el área [8, 13, 21, 22]. La propuesta de uso de nuevas Ontologías como: ESOS y ESO, facilita la explotación de información semántica a usuarios no especializados en el dominio, permitiendo seleccionar para la realización de las consultas, términos equivalentes en dos idiomas (español e inglés). El proceso de enriquecimiento de Ontologías permite transformar el Mundo Cerrado de las Bases de Datos Relacionales al enfoque del Mundo Abierto, facilitando la interoperabilidad entre Bases de Datos, este proceso el cual es llevado a cabo en su generalidad de manera manual, facilita posteriormente la explotación de la información.

Referencias

1. AFUL, “Definición de la Interoperabilidad”. [En línea]. Disponible: <http://interoperability-definition.info/es/>
2. Brickley D., Miller L., “FOAF Vocabulary Specification 0.98”, 09 Agosto 2010. [En línea]. Disponible: <http://xmlns.com/foaf/spec/>
3. Bueno de la Fuente G., “Análisis de la interoperabilidad entre los sistemas de apoyo a la formación de tecminho”, Universidade do Minho, Departamento de Sistemas de Informação. Febrero 2008. [En línea]. Disponible: <http://e-archivo.uc3m.es/bitstream/10016/9089/1/Relatorio%20para%20TecMinho%20v.8.pdf>
4. Cyganiak R., Bizer C., Garbers J., Maresch O., Becker C., “The D2RQ Mapping Language”, 12 Marzo 2012. [En línea]. Disponible: <http://d2rq.org/d2rq-language>
5. Date C.J., “AN INTRODUCTION TO DATABASE SYSTEMS”, in THE SYSTEMS PROGRAMMING SERIES, 1995, Cap.10, pp. 288 – 306.
6. Drummond N., Shearer R., “The Open World Assumption or Sometimes its nice to know what we don’t know”, The University of Manchester. 2006. [En línea]. Disponible: http://www.nesc.ac.uk/talks/701/OWA_NDrummond.pdf
7. Eisenberg V., “D2R Server tutorial”, Computer Science Department, Technion - Israel Institute of Technology, Haifa, Israel. <<http://sw.cs.technion.ac.il/d2rq/tutorial>>.
8. Fink J., Kobsa A., “A review and analysis of commercial user modeling servers for personalization on the world wide web”. User Model. User Adapted Interact. 2000, Cap. 10, pp. 209–249.
9. González Marrón D., “Marco Ontológico para la Interoperabilidad en la Explotación de Información en Bases De Datos Heterogéneas”, Ph.D. Tesis Doctoral, ITESM Campus Estado de México, 2012.

10. Halpin H., “Table of VCard Ontology”. [En línea]. Disponible: <http://www.ibiblio.org/hhalpin/homepage/notes/vcardtable.html>
11. icomputo.com Tecnologías de la información, “Tercera forma normal y la forma normal de Boyce Codd (diseño bases de datos relacionales)”. [En línea]. Disponible: <http://www.icomputo.com/content/tercera-forma-normal-y-la-forma-normal-de-boyce-codd-dise%C3%B1o-bases-de-datos-relacionales>
12. ISWC, “ISWC 2012 – The 11th International Semantic Web Conference”. [En línea]. Disponible: <http://iswc2012.semanticweb.org/>
13. Martínez Villaseñor M. de L., González Mendoza M., y Hernández Gress N., “Towards a Ubiquitous User Model for Profile Sharing and Reuse”, *Sensors* 2012. 28 Septiembre 2012.
14. Microsoft Soporte, “Fundamentos de la normalización de bases de datos”, Id. de artículo: 283878. 22 Febrero 2007. [En línea]. Disponible: <http://support.microsoft.com/kb/283878/es>
15. MySQL hispano, “Normalización de bases de datos”, 29 Mayo 2003. [En línea]. Disponible: <http://www.eet2mdp.edu.ar/alumnos/MATERIAL/MATERIAL/info/infonorma.pdf>
16. Open Data Handbook, “¿Qué son los datos abiertos?”, 2010-2012. [En línea]. Disponible: <http://opendatahandbook.org/es/what-is-open-data/index.html>
17. Oracle and/or its affiliates, “Capítulo 22. La base de datos de información INFORMATION_SCHEMA”, 2011. [En línea]. Disponible: <http://dev.mysql.com/doc/refman/5.0/es/information-schema.html>
18. Ramírez Benavides K., “Ontologías”, 30 Marzo 2005. [En línea]. Disponible: <http://www.slideserve.com/riordan/ontolog-as>
19. Ruble D. A., “EL MODELO DE INFORMACIÓN”, en *Análisis y Diseño Práctico de Sistemas Cliente/Servidor con GUI*, 1998, Cap. 5, pp. 117-159.
20. Sequeda J. F., Hamid Tirmizi S., Corcho O., Miranker D. P., “Survey of directly Mapping SQL databases to the Semantic Web”.
21. Shvaiko P., Euzenat J., “A survey of schema-based matching approaches”. *J. Data Semant.* 2005, Cap. IV, pp. 146–171.
22. Simpson T., Tao T., “WordNet-Based Semantic Similarity Measurement, the Code Project”. [En línea]. Disponible: <http://www.codeproject.com/info/TermsOfUse.aspx/>
23. TechWiki, “Overview of the Open World Assumption”, 5 Julio 2011. [En línea]. Disponible: http://techwiki.openstructs.org/index.php/Overview_of_the_Open_World_Assumption
24. The GNOME Project, “DC Ontology Properties”, 2012. [En línea]. Disponible: <https://developer.gnome.org/ontology/unstable/dc-properties.html>
25. Trinkunas J., Vasilecas O., “Building Ontologies from Relational Databases Using Reverse Engineering Methods”, *International Conference on Computer Systems and Technologies - CompSysTech'07*.
26. W3C Working Group Note, “Manual de SKOS (Simple Knowledge Organization System, Sistema Simple de Organización del Conocimiento)”, 18 Agosto 2009. [En línea]. Disponible: <http://skos.um.es/TR/skos-primer/>

Análisis y evaluación automatizada de modelos paramétricos CAD

Daniel García Calderón, Nestor Velasco Bermeo y Miguel González Mendoza

Escuela de Graduados en Ingeniería y Ciencias (EGIC), ITESM Campus Estado de México,
México

{cdanielgc, nestorvb}@gmail.com,
mgonza@itesm.mx

Resumen En el presente artículo se presenta un Sistema Automatizado de Análisis y Evaluación de Modelos Paramétricos CAD aplicado a los Sistemas de Enfriamiento Automotriz. Se presenta una discusión sobre la importancia de este tipo de aplicaciones en el ámbito industrial y el impacto que tienen los errores de diseño dentro del proceso de desarrollo del producto. La integración permite la evaluación de parámetros de diseño a los sistemas de enfriamiento automotriz con el objetivo de detectar posibles errores en el producto que conllevan a un bajo desempeño o a problemas de calidad. Como resultado de la aplicación de la herramienta, se obtiene un análisis del modelo CAD en tiempo real y se presenta un reporte el cual detalla los resultados obtenidos basados en una evaluación ponderada.

Palabras clave: CAD, sistema de enfriamiento automotriz, modelo paramétrico.

1. Introducción

Actualmente la industria automotriz es una de las más sólidas y crecientes industrias a nivel mundial. De acuerdo al último reporte emitido por la firma Price Waterhouse Coopers, para el año 2017 se espera un crecimiento mundial del 40% en la producción automotriz [1] siendo la tecnología un importante diferenciador. El Diseño Asistido por Computadora [2], el análisis, la evaluación y la optimización juegan un papel muy importante en la generación de mejoras significativas [3]. Sin embargo El diferenciador en la actualidad ya no es de la cantidad de tecnología con la que una empresa cuenta ni del nivel de calidad que manejan sino la forma en la que utilizan la tecnología para disminuir sus tiempos de diseño y producción para integrarse de manera horizontal en la cadena de suministro [4].

El diseño de un producto sigue un proceso como el que se muestra en la Fig. 1. Partiendo de una necesidad que se deriva de las especificaciones de un cliente o del resultado de un estudio de mercado que refleja tal necesidad. Durante el proceso de diseño se concentran dos grandes subprocesos; el de Síntesis y el de Análisis. Durante

el primero de dichos subprocesos se determina la funcionalidad y filosofía por medio de bocetos y dibujos que muestran las relaciones entre las distintas partes del producto. En el caso del subproceso de Análisis se pretende contextualizar el diseño dentro de las ciencias de la ingeniería para evaluar el desempeño del producto.

Lo anterior se consigue gracias al modelado y la simulación. El resultado del proceso de análisis es la documentación del diseño en forma de dibujos de ingeniería

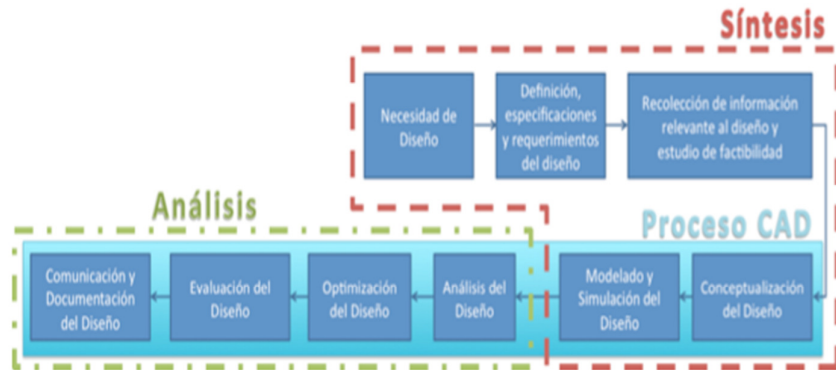


Fig. 1. Diagrama del proceso de diseño de un producto. Tomado de [5] y [6].

Asimismo en la Fig. 1 es posible identificar las distintas etapas dentro del alcance del Diseño Asistido por Computadora (CAD¹) [7] y dentro del cual el producto sufre inevitablemente muchos cambios [8][9] desde su concepción hasta finalmente ser producido. Tales cambios pueden deberse a: errores en la definición de las especificaciones y requerimientos del diseño, durante la creación de los primeros bocetos, defectos de calidad en el producto [10] o cambios en las especificaciones del cliente [11].

En la Fig. 2 se puede apreciar la distribución de los cambios que se dan durante el ciclo de vida de un producto.

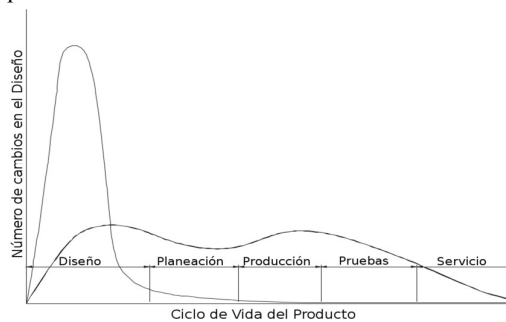


Fig. 2. Cambios en el diseño de un producto durante su ciclo de vida. Tomado de [12].

¹ CAD, Computer Aided Design, por sus siglas en inglés.

De la Fig. 3 se puede comprender la importancia de evitar y prevenir cualquier error que se pueda presentar en la fase de diseño pues a pesar de que solo un 15% del presupuesto se destina a la fase de diseño debido a que el restante 85% ya está comprometido las decisiones que se tomen durante ésta etapa tienen una gran repercusión en términos de costo en el desarrollo del producto.

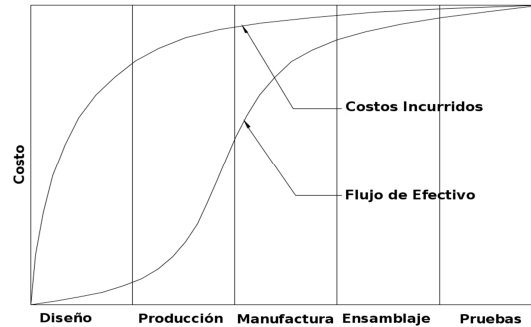


Fig. 3. Distribución de los Costos de Desarrollo de un Producto. Tomado de [12].

Los cambios en el diseño resultan en re trabajo, que se traduce en esfuerzo innecesario al re hacer un proceso o actividad que se llevó a cabo de manera incorrecta la primera vez [13]. Dichos cambios aumentan los costos y tiempo de diseño y sus repercusiones son más evidentes en la etapa de manufactura donde los cambios de procesamiento absorben de un 20% a un 50% la capacidad de desarrollo de un producto [14].

Una técnica comúnmente usada para asegurar la homogeneidad en familias de partes en la fase de diseño es haciendo uso de modelos CAD paramétricos. Un modelo paramétrico se caracteriza por basarse en Fig.s geométricas que obedecen a una serie de restricciones que definen: sus dimensiones, posición en el espacio, posición con relación a otra pieza, etc.

Tomando como base una geométrica plana con forma de “L”. Si se tiene una Fig. geométrica cerrada con dimensiones: d_1 , d_2 , d_3 , d_4 respectivamente y se ha definido que la dimensión d_1 debe ser mayor a la dimensión d_2 tenemos que dicha restricción se representa matemáticamente como $d_2 > d_1$. Lo anterior se puede ver en la Fig. 4. De tal restricción es posible que se obtenga una Fig. que cumpla con tales restricciones (a) y una Fig. que no cumpla con tales restricciones y el resultado sea una geometría no válida (b).

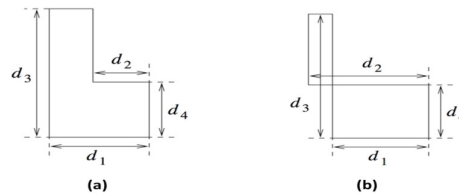


Fig. 4. Parametrización de una Fig. geométrica. Tomado de [16].

Siguiendo la misma restricción antes definida ($d_2 > d_1$). Es posible observar en la Fig. 5 un par de geometrías con una parametrización válida (a) y otras geometrías incorrectamente parametrizadas y por consiguiente generan diseños inválidos a pesar de que se cumple con la restricción antes mencionada.

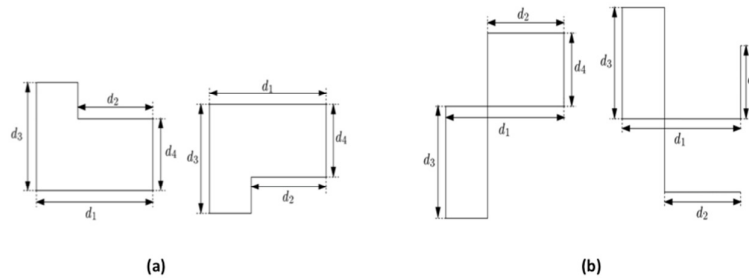


Fig. 5. Parametrización correcta (a) y Parametrización incorrecta (b). Tomado de [16].

Es evidente que a pesar de que un modelo CAD ha sido parametrizado es posible que se obtenga una geometría inválida o que el modelo ha sido pobremente parametrizado, la probabilidad de que existan errores en el diseño del producto se incrementan. Actualmente ningún sistema CAD es capaz de analizar y verificar que la geometría resultante cumpla con valores definidos por la empresa y que están alineados a una política de calidad. La validación del cumplimiento de las restricciones de diseño es comúnmente hecho por el diseñador y registrado en un documento de papel (checklist) debido a la carencia antes descrita.

Es por tal motivo que la presente investigación tiene como objetivo el desarrollar un sistema que permite el Análisis y Evaluación de modelos paramétricos CAD dentro de un ambiente de Diseño Computarizado. Tomando como base el análisis de modelos y geometrías para la búsqueda de inconsistencias entre el modelo geométrico y un conjunto de parámetros definidos por la empresa y su política de calidad correspondiente a cada producto, con el objetivo de identificar y corregir los errores que se presenten asegurando así que el diseño cumpla en su totalidad con tales estándares y restricciones.

El presente artículo se organiza de la siguiente manera: la Sección 2 presenta los trabajos relacionados con la evaluación de modelos paramétricos, en la Sección 3 se detalla el caso de uso para el cual el sistema está construido y los objetivos que se han definido como base para el desarrollo, en la Sección 4 se presentan los detalles de funcionamiento y operación del sistema propuesto, y por último en la Sección 5 se presentan y discuten los resultados obtenidos a la par del trabajo futuro.

2. Trabajos relacionados

Como se mencionó anteriormente, durante el proceso de diseño de un producto se pueden incurrir en una serie de errores que posteriormente tienen repercusiones considerables elevando los costos de producción e incrementando el tiempo de producción.

El trabajo de Möhringer [15] enfatiza en el hecho de que cada error incorpora oportunidades de mejorar el producto y el proceso de desarrollo del mismo. Define un proceso compuesto de seis pasos: Evitar errores, registrar los errores, clasificar los errores, resolver los errores, evaluar los errores y proporcionar conocimiento.

Cada una de las etapas tiene como base el ingreso de información y como salida principal es la documentación apropiada dependiendo del error el cual se está refiriendo. Ahora bien en términos de análisis y verificación de restricciones en modelos paramétricos CAD se observan dos grandes líneas de trabajo, por un lado están los trabajos que atacan el problema con modelos en dos dimensiones (2D) y por el otro lado las aplicaciones enfocadas a modelos en tres dimensiones (3D).

2.1. Aplicaciones desarrolladas para modelos en 2D

La validación de modelos paramétricos con el fin de obtener figuras geométricas válidas es descrita por Hoffman [16] basándose en la confirmación de las restricciones horizontales y verticales para una figura de dos dimensiones (2D), obteniendo el rango de las variaciones para ambas restricciones sin cambiar la topología original. Cada figura es representada por un árbol y así establece si el resultado obtenido es válido, dependiendo de los árboles validos se determinan los valores que se pueden ocupar para construir un polígono válido.

Asimismo Zhang [19] aborda el problema de modelos geométricos 2D pobremente parametrizados en términos de soluciones y con figuraciones no degenerativas. Partiendo de un modelo que no ha sido debidamente parametrizado (pobremente restringido) desarrolló un algoritmo que permite automáticamente agregar nuevas restricciones en el grafo del diseño. Las nuevas restricciones permiten que el diseño resultante esté bien parametrizado y la serie de ecuaciones que de tales restricciones se derivan pueden ser resueltas de manera simultánea y no superan el grado de las restricciones originales.

2.2. Aplicaciones desarrolladas para modelos en 3D

En términos de validación y verificación de modelos en tres dimensiones (3D) Li [18] presenta un método para construir grafos basados en los Grados De Libertad (GDL) de la parametrización de un diseño. Basándose en el principio de que es posible reducir el número de variables paramétricas al agruparlas (clusters) de acuerdo al análisis de las relaciones estructurales entre las restricciones y el modelo geométrico. Una vez que se ha hecho de manera recursiva la creación de clusters de parámetros basado en el grafo de los GDL del diseño 3D se resuelve el modelo y se reduce el número de restricciones para obtener un modelo válido.

El trabajo de Wallner [18] se enfoca en la propagación de errores debido a errores en la parametrización geométrica de una figura. Para manejar la propagación de los errores propone un método que se enfoca primordialmente en determinar zonas de tolerancia por medio de restricciones implícitas en una Fig. geométrica y partiendo del hecho de objeto geométricos definidos de forma imprecisa pero de una zona de tolerancia determinada haciendo énfasis en restricciones relacionadas con su posición.

Los trabajos antes descritos presentan aplicaciones y metodologías se enfocan primordialmente en el análisis y definición de las variantes, posibles valores y umbrales de los valores que pueden tomar las distintas restricciones paramétricas de un modelo geométrico.

Sin embargo ninguno presenta evidencia de una metodología que analice y evalué los valores y reglas de parametrización de un modelo geométrico en un ambiente de Diseño CAD para prevenir errores de diseño.

3. Caso de uso: sistemas de enfriamiento automotriz

Un Sistema de Enfriamiento Automotriz (SEA) es un componente de gran importancia en la industria automotriz y de gran relevancia en el desempeño en los motores de combustión interna pues proporciona el flujo de aire necesario para mantener el automóvil a una adecuada temperatura de trabajo. Dentro de los componentes principales que conforman un SEA podemos encontrar: la tolva, el ventilador y el motor. Un ejemplo de ensamblaje y las relaciones que obedecen entre cada una de ellas es posible de ver en la Fig. 6.

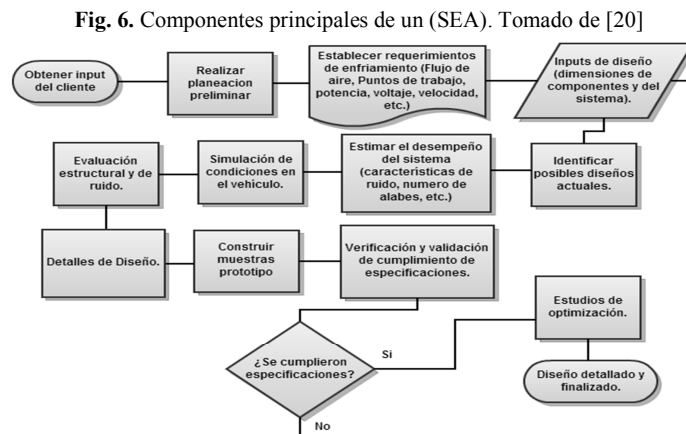
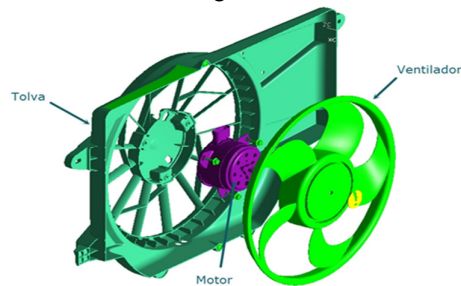


Fig. 7. Diagrama de proceso de diseño de Sistema de Enfriamiento Automotriz

Una de las partes más críticas para el diseño de los SEA (Fig. 7) es el ventilador pues es la parte que proporciona la cantidad de aire suficiente para mantener el automóvil a la temperatura de trabajo.

La tolva es parte esencial del desempeño del SEA y de ahí que el diseño en conjunto van de la mano para poder trabajar de manera eficiente y sin problemas o errores de funcionamiento. El motor es el corazón del sistema ya que es el que impulsa al ventilador a una velocidad específica, sin embargo el diseño del motor no es considerado dentro del presente trabajo. En términos de las relaciones y restricciones que existe entre cada uno de los elementos que componen un SEA es imperativo que las reglas de parametrización sean satisfechas en su totalidad debido a que si alguna restricción es violada el diseño del SEA en su totalidad conlleva a problemas en su ensamblaje pero más importante aún el desempeño dentro de un automóvil es nulo.

4. Sistema automatizado de análisis y evaluación de modelos paramétricos CAD

Algunos de los requerimientos que se definen al momento de diseñar los componente de un SEA se destaca la cantidad de flujo de aire y consumo de corriente, pero para el diseño mecánico es de suma importancia lograr un empaquetamiento con el motor de combustión (área disponible para el ensamblaje de los sistemas) de aquí que cada sistema es único y exclusivo de cada vehículo, por ello su diversidad en variantes de diseño. El sistema de evaluación tiene la capacidad de realizar un análisis generalizado de estos sistemas bajo las condiciones en las cuales debieron ser creadas. Se realiza una verificación de parámetros basadas en espacios o distancias entre componentes, los cuales representan puntos críticos para el buen desempeño del sistema.

Este sistema fue desarrollado dentro de la estructura de operación del software de diseño CATIA². El sistema desarrollado se divide en 4 módulos principales que realizan las operaciones para la evaluación de los modelos CAD: Módulo Principal, Módulo de Interacción con Usuarios, Módulo de Evaluación de Parámetros y Módulo de Generación de Reportes y Resultados. Una explicación detallada de cada uno de estos módulos se presenta a continuación.

4.1. Módulo principal (MP)

Este módulo es el encargado de hacer las llamadas a los módulos para su ejecución. Es el principal sistema de toma de decisiones y el cual tiene como principal objetivo de controlar el flujo de operaciones y comunicación entre los demás módulos para la correcta realización de las tareas de la herramienta de evaluación.

² Dassault Systems, CATIA. <http://www.3ds.com/products/catia/>

4.2. Módulo de interacción con usuarios (MIU)

El módulo de interacción con usuarios tiene el principal objetivo de manejar todas las operaciones que tienen que ser manejadas con el usuario. Este módulo contiene el código con el que presenta diversas formas en las cuales solicita información o le pide al usuario tomar decisiones.

Las operaciones que realiza son: Selección del tipo de evaluación a realizar, ubicación de archivo a abrir para el análisis, input de datos del SEA como parámetros de generalización, etc. El input de datos contempla el posicionamiento del modelo 3D dentro del automóvil, diámetro del ventilador, tipo de motor, entre otros.

4.3. Módulo de evaluación de parámetros (MEP)

Para poder realizar la evaluación del modelo CAD se generan secciones del modelo en 2D definidas como “regiones de interés”, sobre las cuales se lleva a cabo el dimensionamiento de los elementos geométricos contenidos en dicha región para su posterior evaluación. Dentro de estas regiones de interés se asocian la mayor cantidad de restricciones y así se minimiza el número de evaluaciones a realizar dentro de las mismas. Un ejemplo de las regiones de interés derivada de un diseño puede observarse en la Fig. 8.

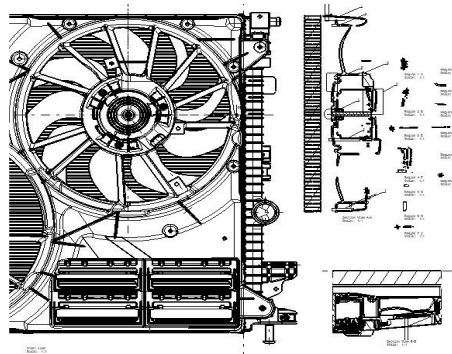


Fig. 8. Corte de sección y regiones de interés.

Este módulo toma las dimensiones generadas y les asigna una ponderación. Esta ponderación es de acuerdo al grado de severidad en el ensamble final si este parámetro estuviera fuera de rango. Se comparan los valores obtenidos. Al generarse dicha calificación, se realiza una llamada al módulo de generación de reportes y resultados para asignar al parámetro analizado la evaluación correspondiente y registrar este resultado en el archivo del reporte del diseño. Es importante señalar que los parámetros evaluados son tomados de acuerdo a reglas de diseño establecidas en el departamento de diseño de sistemas térmicos de la empresa Robert Bosch.

4.4. Módulo de generación de reportes y resultados (MGRR)

Este módulo se encarga de realizar las actividades de generación de resultados y asignarlos a un archivo de Excel, que posteriormente se convertirá en el reporte de la evaluación del modelo. Sus principales tareas constan en registrar en un archivo de Excel los datos derivados de la evaluación realizada en el MEP siguiendo una estructura de hoja de trabajo y por último presentar el reporte final al usuario. Fig. 9.

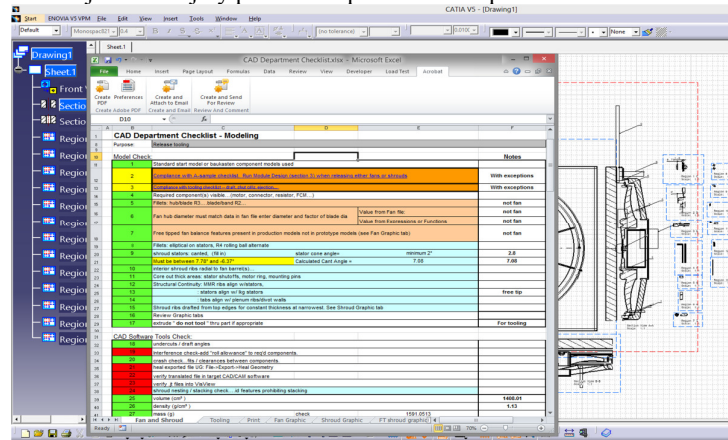


Fig. 9. Reporte de evaluación generado automáticamente.

La Fig. 10 presenta un esquema general de la interconexión y comunicación entre el software de diseño (CATIA) y el módulo interconectado con Excel para generar los reportes.

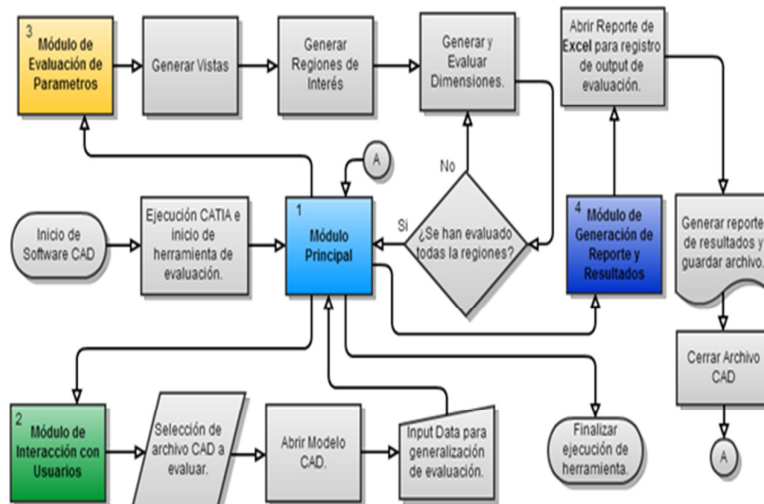


Fig. 10. Diagrama de interconexión y comunicación de módulos.

5. Discusión y trabajo futuro

Para comprobar el correcto análisis y evaluación de modelos paramétricos CAD, se evaluaron una serie de 30 diseños del diseño denominado “Ventilador con Banda”. Dentro de los parámetros más críticos dentro del diseño se tiene el ancho de la banda, el grosor del eje y diámetro del ventilador. De los 30 diseños analizados fue posible encontrar un 90% de los errores en los parámetros registrados en los diseños contra los valores definidos para que se acepte el modelo paramétrico CAD. El sistema aquí presentado se realizó basado en los sistemas de enfriamiento automotriz, sin embargo, esto no es un impedimento de implementarse en otro tipo de producto.

Referencias

1. Price Waterhouse Coopers: Consolidation in the Global Automotive Supply Industry 2012, PWC (2012)
2. Bordegoni, M., Rizzi, C.: Innovation in Product Design: From CAD to Virtual Prototyping. London: Springer (2011)
3. Girardot, M., y Schwarz M.: Electronics-ization in Automotive, Reinventing the Industry Model To Boost Profitable Innovation , Cisco Internet Business Solutions Group (IBSG) (2012)
4. Rishi, S., Stanley, B., & Gyimesi, K.: Automotive 2020: Clarity beyond the chaos. IBM Institute for Business Value. August (2008)
5. Zeid, I.: Mastering cad/cam. McGraw-Hill, Inc. (2004)
6. Rao, P. N.: CAD/CAM Principles and Applications (Sc). Tata McGraw-Hill Education (2010)
7. Groover, M., & Zimmers, E. W. J. R.,1983. CAD/CAM: computer-aided design and manufacturing. Prentice Hall.
8. Hanna, A.S. y Swanson, J.: Risk allocation by law-cumulative impact of change orders. Journal of Professional Issues in Engineering Education and Practice, ASCE (2007)
9. Kohler, C., et al.: A matrix representation of the CPM/PDD approach as a means for change impact analysis. Conferencia Internacional de Diseño, Dubrovnik, Croacia (2008)
10. Burati, J.L., Farrington, J.J., y Ledbetter, W.B.: Causes of quality deviations in design and construction. Journal of Construction Engineering and Management (1992)
11. Keller, R., Eckert, C.M., and Clarkson, P.J.: Using an engineering change methodology to support conceptual design. Journal of Engineering Design, 20(6) (2009)
12. Radhakrishnan, P., Subramanyan, S., & Raju, V.: CAD/CAM/CIM. New Age International (2008)
13. Love, P. E.: Influence of project type and procurement method on rework costs in building construction projects. Journal of construction engineering and management, 128(1) (2002)
14. Ehrlenspiel, K., Kiewert, A., and Lindemann, U.: Cost-efficient design. Berlin, Germany: Springer-Verlag (2007)
15. Mohringer, S.: From design errors to chances – a computer-based error tracking system in practice. Conferencia Internacional de Diseño, Dubrovnik, Croacia (2008)
16. Hoffmann, C. M., & Kim, K. J.: Towards valid parametric CAD models. Computer-Aided Design, 33(1), 81-90 (2001)

17. Zhang, G. F.: Well-constrained completion for under-constrained geometric constraint problem based on connectivity analysis of graph. In: Proceedings of the 2011 ACM Symposium on Applied Computing, pp. 1094-1099, ACM (2011)
18. Li, Y. T., Hu, S. M., & Sun, J. G.: A constructive approach to solving 3-D geometric constraint systems using dependence analysis. *Computer-Aided Design*, 34(2), 97-108 (2002)
19. Wallner, J., Schröcker, H. P., & Hu, S. M.: Tolerances in geometric constraint problems. *Reliable computing*, 11(3), 235-251 (2005)
20. Robert Bosch Corporation, SEA System Design.

Distribución balanceada de ayuda humanitaria resuelto con un algoritmo metaheurístico

Ivo Erasmo Buzón Cantera¹, Jaime Mora Vargas¹,
Ángel Ruiz Bartolomé² y Patrick Soriano³

¹ Tecnológico de Monterrey Campus Estado de México, México
milvus_ca@yahoo.com, jmora@itesm.mx

² Université Laval, Quebec, Quebec, Canadá
angel.ruiz@fsa.ulaval.ca

³ HEC Montréal, Montreal, Quebec, Canadá
patrick.soriano@hec.ca

Resumen La distribución de ayuda humanitaria tras un desastre busca satisfacer las necesidades básicas de los afectados. La mayor dificultad es la planificación de la distribución desde el centro de distribución local a los puntos de distribución, donde la ayuda humanitaria es repartida directamente a los afectados. La falta de ayuda humanitaria, la carencia de capacidad de transporte y una red de comunicación dañada son situaciones que tienen que hacer frente los administradores de la distribución de ayuda humanitaria. Este trabajo desarrolla un modelo de la distribución balanceada de ayuda humanitaria basado en el modelo del problema de ruteo. Para resolver el problema de distribución balanceada de ayuda humanitaria este trabajo desarrolla un algoritmo híbrido entre un algoritmo metaheurístico y un método exacto. Los resultados experimentales muestran que el algoritmo propuesto se desempeña bien para el problema de distribución balanceada de ayuda humanitaria.

Palabras clave: metaheurísticas, distribución balanceada, ayuda humanitaria.

1. Introducción

Los desastres de origen natural han afectado negativamente a la humanidad. Un desastre produce destrucción, pérdidas humanas y materiales, dejando a una población desvalida que no puede satisfacer sus necesidades básicas. Las infraestructuras han sido parcial o totalmente destruidas y las cadenas de suministros habituales están inoperantes [11]. Dentro de la logística humanitaria, la distribución de ayuda humanitaria (DAH) es la actividad que realiza el transporte de la ayuda humanitaria desde el centro de distribución local (CDL) a los puntos de distribución (PD) mediante una flota heterogénea de vehículos. Desde los puntos de distribución la ayuda humanitaria es repartida directamente a la población afectada. Balcik et al. [2] establecen que un mal desempeño de la DAH produce sufrimiento a los afectados por una privación de sus necesidades básicas. Kovacs y Spens [10] indican que el atraso en el suministro de ayuda humanitaria puede resultar en pérdida de vidas. En

consecuencia, un buen plan de la DAH es absolutamente necesario para repartir la ayuda humanitaria disponible. Huang et al. [8] y Perez et al. [13] resaltan que el plan de DAH es realizado frecuentemente de manera ad-hoc y puede llevar a un uso ineficiente de los recursos, a una respuesta lenta y a una injusta distribución de la ayuda. De manera similar Ergun et al. [7] detectan que las personas responsables de la administración de la cadena de suministro y de la logística humanitaria en la mayoría de las organizaciones generalmente no son especialistas, y no tienen las herramientas o sistemas adecuados para resolver los problemas relacionados con la DAH. Una respuesta exitosa a un desastre no es improvisada, Van Wassenhove [18] recomienda estar preparados para establecer la respuesta más efectiva. Pero la realización del plan de DAH de manera rápida y óptima es una tarea difícil. Las técnicas de investigación de operaciones aplicadas a la DAH pueden mejorar su eficiencia y eficacia enormemente [2, 7]. La distribución de ayuda humanitaria es diferente a otros problemas de distribución. El objetivo de la distribución humanitaria es proveer asistencia humanitaria, no la minimización de los costos. Un problema que tienen que hacer frente los administradores es la limitación de recursos, representada por la ayuda humanitaria disponible y la capacidad de transporte [2, 7, 10].

En este trabajo, se desarrolla un modelo del problema de DAH que incorpora el balanceo de la distribución. El modelo incluye las decisiones de ruteo y las decisiones de asignación de ayuda humanitaria a los PD. El modelo se desarrolla a partir de la formulación de dos índices del problema de ruteo de vehículos [17]. El modelo se formula con múltiples periodos para poder considerar adecuadamente la realización de una distribución balanceada, la satisfacción de la demanda con atrasos y el inventario en los puntos de distribución. Para resolver el problema de DAH se adapta el algoritmo metaheurístico de recocido simulado definiendo una función perturbación con múltiples vecindades. También se desarrolla un algoritmo mateheurístico que tiene como algoritmo principal un algoritmo de recocido simulado que incluye la resolución de un subproblema con programación entera para explorar una gran vecindad.

2. Estado del arte

El trabajo de Altay y Green III [1] es la única revisión de la literatura en administración de la logística humanitaria. Concluyen que existe poca investigación sobre la planificación del transporte para la distribución de la ayuda humanitaria. Balcik et al. [2] y Sheu [16] identifican oportunidades para el uso de sistemas inteligentes de transporte en la DAH. En los trabajos presentados se presentan diferentes modelos de la DAH, cada trabajo presenta el problema de DAH de manera diferente en función de las decisiones que quiere responder y de los supuestos establecidos. La resolución del problema de DAH mediante métodos exactos solo puede ser realizada para instancias de pequeño tamaño en un tiempo razonable [2, 19], sin embargo De Angelis et al. [5] resuelven el modelo propuesto con programación entera mixta. Por la complejidad del problema, los métodos presentados descomponen el problema en dos etapas [2, 3, 11, 19], aunque existen varios trabajos que resuelven el problema en una sola etapa [5, 12]. Barbarosoglu et al. [3] presentan un método heurístico interactivo para la coordinación de los dos sub

problemas con un sistema de múltiples criterios jerárquico. Yi y Ozdamar [19] y Perez et al. [13] utilizan un método con dos etapas: La primera etapa define un problema de flujo en redes entero mixto, en el cual los vehículos se representan como productos siendo su flujo una variable entera. La segunda etapa define las rutas y la carga para cada vehículo. Balcik et al. [2] presentan un sistema con dos fases. La primera fase genera todas las rutas no dominadas. La segunda fase mediante programación entera mixta asigna las rutas a los vehículos, determina la carga de los vehículos y las cantidades a distribuir en cada PD. Lin et al. [11] presentan dos heurísticos. El primer heurístico reduce el número de rutas al considerar un subconjunto, las rutas se filtran mediante un algoritmo genético. El segundo heurístico descompone el problema inicial en subproblemas que solo contienen un número parcial de vehículos y de PD. Nolz et al. [12] presentan una formulación matemática entera no lineal y un método heurístico basado en el algoritmo genético de ordenamiento no-dominado II (NSGA-II). En Yi y Ozdamar [19] el objetivo es minimizar la demanda de ayuda no satisfecha y Perez et al. [13] utilizan el concepto de costo social, estableciendo el objetivo como minimizar la suma de los costos operacionales y los costos sociales. Huang et al. [8] presenta un modelo de la DAH para un solo periodo sin limitaciones, el objetivo minimiza el balance en el tiempo de reparto a los PD.

3. Descripción del problema

El problema de distribución balanceada de ayuda humanitaria (PDBAH) puede ser formalmente definido como sigue. Sea $G=(V,A)$ un grafo dirigido donde $V=\{0, \dots, n\}$ es el conjunto de los nodos y $A=\{(i,j):i,j \in V, i \neq j\}$ es el conjunto de los arcos. El nodo 0 corresponde al CDL y el resto de nodos $\{1, \dots, n\}$ representan los PD. Un tiempo de recorrido c_{ij} no negativo está asociado a cada arco $(i,j) \in A$ y representa el tiempo necesario para recorrer la distancia del nodo i al nodo j . En el presente trabajo se considera que los tiempos de recorrido son simétricos, es decir, $c_{ij} = c_{ji}$ para todo arco $(i,j) \in A$. Una flota heterogénea está disponible, representada por el conjunto k , y cada vehículo está limitado en la capacidad de carga (CAP_k) y en el tiempo de trabajo máximo (W). Cada PD i , ($i=1, \dots, n$) está asociado con una conocida demanda d_{it} no negativa para cada periodo t del horizonte de planeación, el CDL está asociado a una demanda igual a cero. La ayuda disponible en el CDL es conocida y no negativa para cada periodo t del horizonte de planeación e inferior en cada periodo a la demanda total. Una representación esquemática del PDBAH se presenta en la figura 1. La solución considera también la capacidad de almacenaje de ayuda humanitaria en los PD para satisfacer la demanda del periodo siguiente, que los vehículos puedan realizar varias rutas en cada periodo, varios vehículos pueden visitar el mismo PD en un mismo periodo, se permite satisfacer parcialmente la demanda de los PD y los vehículos pueden realizar varias rutas sin sobrepasar el tiempo de trabajo máximo. El objetivo principal del PDBAH es minimizar las penalizaciones por satisfacer la demanda tardíamente. Los factores de penalización se definen en relación con el atraso: a) Si la demanda es satisfecha en el mismo periodo no hay penalización. b) Si la demanda es satisfecha con un atraso de 1 o 2 periodos se define el factor de penalización p_1 y p_2 . c) Si la demanda no es satisfecha está asociada a un factor de

penalización p_3 . También incluye un objetivo secundario que es el balance de la distribución entre los PD. El balance busca un grado de equidad en el servicio recibido por los PD. La equidad total que todos reciban exactamente el mismo servicio, no puede lograrse por las limitaciones en los recursos del problema. El servicio recibido por cada PD se mide al calcular la demanda no satisfecha porcentual. El balance del servicio se incorpora al objetivo al incluir la demanda no satisfecha porcentual de cada PD, a través de una función convexa lineal por segmentos que relaciona el porcentaje de demanda no satisfecha del PD con un peso. Huang et al. [8] presentan que esta forma de establecer el balance tiene dos ventajas frente a minimizar la máxima diferencia del servicio recibido por los PD. La primera ventaja es que el modelo se mantiene lineal. La segunda ventaja es que se obtienen mejores resultados en los otros objetivos. La solución del PDBAH establece: Los PD que reciben ayuda humanitaria en cada periodo, La cantidad de ayuda humanitaria que recibe cada PD en cada periodo, Las rutas de los vehículos y la asignación de las rutas a los vehículos.

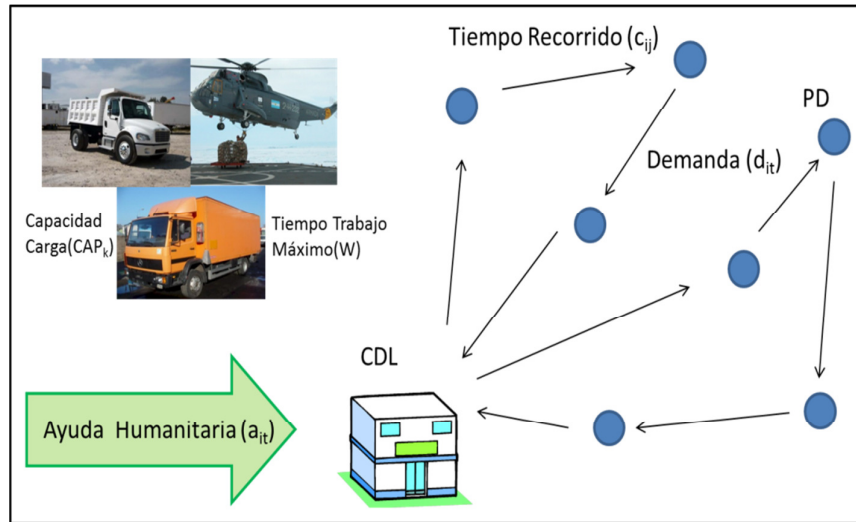


Fig. 1. Representación esquemática del PDBAH.

3.1. Formulación del problema

La siguiente nomenclatura es utilizada para formular el PDBAH:

- I,J: conjunto de nodos
- T: conjunto de periodos de tiempo, tamaño del horizonte de planeación
- K: conjunto de vehículos
- R: conjunto de rutas
- B: segmentos de la función convexa lineal por segmentos

CAP_k : capacidad del vehículo k (unidades de producto)

W : tiempo de trabajo máximo del vehículo k

p_1, p_2 y p_3 : factores de penalización por atraso

d_{it} : demanda del PD i en el periodo t

a_t : la ayuda total disponible a ser repartida en el periodo t

c_{ij} : tiempo de recorrido del nodo i al nodo j

M : es un valor muy grande

$xval_b, yval_b$: valores en las intersecciones de la función convexa lineal por segmentos, $f(xval_b) = yval_b$

x_{ijrkt} : 1 si el vehículo k viaja del nodo i al nodo j en el periodo t en la ruta r , 0 en otro caso.

y_{jrkt} : 1 si el vehículo k visita al PD j en el periodo t en la ruta r , 0 en otro caso.

s_{ikrt} : cantidad de ayuda a dar por el vehículo k en la ruta r en el periodo t en el PD i para la demanda del t .

inv_{ikrt} : cantidad de ayuda dará el veh. k en la ruta r en el periodo t al PD i para la demanda del periodo $t+1$.

$da1_{ikrt}$: cantidad de ayuda dará el veh. k en la ruta r en el periodo t al PD i para la demanda del periodo $t-1$.

$da2_{ikrt}$: cantidad de ayuda dará el veh. k en la ruta r en el periodo t al PD i para la demanda del periodo $t-2$.

dn_{it} : cantidad de demanda no satisfecha del PD i del periodo t .

pdn_i : porcentaje de la demanda no satisfecha del PD i

$pesopd_i$: peso de la demanda no satisfecha del PD i

pos_b : variable para establecer la relación entre pdn_i y $pesopd_i$ con la función convexa lineal por segmentos.

El modelo del PDBAH:

$$\text{Min} \left[\sum_{i \in V \setminus \{0\}} \sum_{t \in T} \left(\sum_{k \in K} \sum_{r \in R} p_1 da1_{ikrt} + p_2 da2_{ikrt} \right) + p_3 dn_{it} \right] + \sum_{i \in V \setminus \{0\}} pesopdn_i \quad (1)$$

Sujeto a:

$$\sum_{j \in V} x_{ijrkt} = \sum_{j \in V} x_{jirkt} = y_{irkt}, \forall (i, r, k, t) \quad (2)$$

$$\sum_{i \in V} \sum_{j \in V} \sum_{r \in R} c_{ij} x_{jirkt} \leq W, \forall (k, t) \quad (3)$$

$$\sum_{i \in V} da2_{ikrt} + da1_{ikrt} + s_{ikrt} + inv_{ikrt} \leq CAP_k, \forall (r, k, t) \quad (4)$$

$$\sum_{i \in V} \sum_{r \in R} \sum_{k \in K} da2_{ikrt} + da1_{ikrt} + s_{ikrt} + inv_{ikrt} \leq a_t, \forall (t) \quad (5)$$

$$da2_{irkt} + da1_{irkt} + s_{irkt} + inv_{irkt} \leq M y_{irkt}, \forall(i, r, k, t) \quad (6)$$

$$\sum_{k \in K} \sum_{r \in R} inv_{irkt} \leq d_{it+1}, \forall(i, t = 1, \dots, T - 1) \quad (7)$$

$$inv_{irkt} = 0, \forall(i, r, k, t = T) \quad (8)$$

$$d_{it} \leq \sum_{k \in K} \sum_{r \in R} s_{irkt} + da1_{irk,t+1} + da2_{irk,t+2} + dn_{it}, \forall(i, t = 1) \quad (9)$$

$$d_{it} \leq \sum_{k \in K} \sum_{r \in R} inv_{irk,t-1} + s_{irkt} + da1_{irk,t+1} + da2_{irk,t+2} + dn_{it}, \forall(i, t = 2, \dots, T - 2) \quad (10)$$

$$d_{it} \leq \sum_{k \in K} \sum_{r \in R} inv_{irk,t-1} + s_{irkt} + da1_{irk,t+1} + dn_{it}, \forall(i, t = T - 1) \quad (11)$$

$$d_{it} \leq \sum_{k \in K} \sum_{r \in R} inv_{irk,t-1} + s_{irkt} + dn_{it}, \forall(i, t = T) \quad (12)$$

$$u_{ikrt} - u_{jkrt} + CAP_k x_{ijrkt} \leq CAP_k - d_{jt}, \forall(i, j, k, r, t) \quad (13)$$

$$d_{it} \leq u_{irkt} \leq CAP_k, \forall(i, r, k, t) \quad (14)$$

$$pdn_i = \frac{\sum_{t \in T} dn_{it}}{\sum_{t \in T} d_{it}}, \forall i \quad (15)$$

$$pdn_i = \sum_{b \in B} pos_{ib} xval_b, \forall i \quad (16)$$

$$pesopdn_i = \sum_{b \in B} pos_{ib} yval_b, \forall i \quad (17)$$

$$\sum_{b \in B} pos_{ib} = 1, \forall i \quad (18)$$

$$x_{ijrkt} \in \{0,1\}, \forall(i, j, r, k, t) \quad (19)$$

$$y_{irkt} \in \{0,1\}, \forall(i, r, k, t) \quad (20)$$

$$s_{irkt}, da1_{irkt}, da2_{irkt}, inv_{irkt} \geq 0, \forall(i, r, k, t) \quad (21)$$

$$dn_{it} \geq 0, \forall(i, t) \quad (22)$$

$$pdn_i, pesopdn_i, pos_{ib} \in (0,1), \forall(i, b) \quad (23)$$

La función objetivo (1) busca minimizar la penalización de los atrasos como objetivo principal y el objetivo secundario es la realización de una distribución balanceada. La restricción (2) establece la conservación de flujo y la asociación vehículo - PD - periodo - ruta. La restricción (3) permite considerar el tiempo de trabajo máximo. De manera similar, la restricción (4) considera la capacidad de carga del vehículo. La restricción (5) considera la cantidad de ayuda humanitaria disponible. La restricción (6) tiene en cuenta que se entreguen cantidades si el PD es visitado. Las restricciones (7 y 8) limitan los valores máximos de inventario. Restricciones (9,

10, 11 y 12) asocian la demanda con la ayuda humanitaria entregada y la demanda no satisfecha. La restricción (13 y 14) permiten eliminar los subtours. La restricción (15) calcula el valor del porcentaje de la demanda no satisfecha. La restricción (16) relaciona el porcentaje de la demanda no satisfecha a través de la función convexa lineal por segmentos para establecer el valor de $posib$. La restricción (17) calcula el peso del servicio por demanda no satisfecha $pesopdi$ a través de la función convexa lineal por segmentos y el valor de $posib$. La restricción (18) establece que $posib$ tiene un valor de 1, para establecer adecuadamente el valor de $pesopdi$ a partir de $pdni$. Las ecuaciones (19 - 23) limitan a las variables de manera general. Analizando la formulación del PDBAH se observa que es la unión de dos problemas: el problema de asignación de ayuda humanitaria y el problema de ruteo.

4. Algoritmo recocido simulado

Existen diferentes caminos para solucionar problemas de optimización combinatorios como el PDBAH. Raidl y Puchinger [15] clasifican en dos categorías las técnicas para resolver los problemas de optimización combinatorios: los métodos de programación matemática o métodos exactos y los métodos heurísticos o metaheurísticos. Entre los métodos metaheurísticos, Kirkpatrick et al. [9] introdujeron hace poco más de 25 años el algoritmo recocido simulado que ha demostrado ser una herramienta exitosa para resolver una amplia gama de problemas de optimización combinatoria. Su nombre proviene del proceso de enfriamiento de materiales: recocido [6]. El recocido simulado es el más antiguo entre los metaheurísticos y uno de los primeros algoritmos que tienen una estrategia explícita para salir de los óptimos locales. La idea fundamental es permitir movimientos que resultan en soluciones de peor calidad que la solución actual. Este movimiento tiene el propósito de escapar del óptimo local actual para poder explorar el espacio de soluciones y eventualmente encontrar otro óptimo local de mejor calidad. La probabilidad de hacer tal movimiento disminuye durante al búsqueda.

La figura 2 presenta el pseudo código del algoritmo de recocido simulado para problemas de maximización con función de costo c . El algoritmo empieza generando una solución inicial, que pasa a ser la solución actual s . La solución inicial se genera generalmente de forma aleatoria o heurística. Luego se inicializan los parámetros: La variable temperatura $Temp$ toma el valor de la temperatura inicial $Temp_{ini}$, el factor de reducción de la temperatura α , las repeticiones del ciclo interno $iter$ y la temperatura final o criterio de paro $Temp_{fin}$.

Fig. 2. pseudo código del algoritmo de recocido simulado.

El algoritmo de recocido simulado consta de dos ciclos, uno externo (o ciclo de temperatura) y uno interno (o ciclo Metropolis). El siguiente paso inicia el ciclo externo que termina cuando la temperatura $Temp$ sea inferior al criterio de paro $Temp_{fin}$. El núcleo del algoritmo es el ciclo interno que se repite $iter$ veces. En el ciclo interno, primero se establece una nueva solución s_{new} diferente de la solución actual s , mediante una función perturbación o mecanismo de vecindad. La solución s_{new} es una solución de la vecindad de s . Se calcula el incremento de energía ΔE como la diferencia entre el costo de la nueva solución $c(s_{new})$ y el costo de la solución actual

$c(s)$. Si $\Delta E > 0$ entonces la solución nueva s_{new} es mejor que la solución actual s . La nueva solución s_{new} se convierte en la solución actual s . Sin embargo, a diferencia de los algoritmos de búsqueda local, si $\Delta E < 0$ entonces se genera un número aleatorio y se compara con $\exp(\Delta E / Temp)$. De ser menor, entonces la nueva solución s_{new} se convierte en la solución actual s . El número aleatorio se genera según una función de probabilidad uniforme entre 0 y 1. Es importante observar que mientras disminuye la temperatura disminuye la probabilidad de realizar cambios que impliquen un empeoramiento del costo de la solución actual s . Tras terminar el ciclo interno el valor de la temperatura $Temp$ es actualizado según el parámetro α . El valor de α debe ser inferior a 1 para que la temperatura disminuya.

Paso 1: Inicialización

s = solución inicial
temp = Temperatura inicial
 α = factor de reducción
iter = número de iteraciones
 T_{fin} = Temperatura final (criterio de paro)

Paso 2: Búsqueda

mientras Temp > T_{fin}
desde $x=1$ a iter

Paso 3: Nueva solución

s_{new} = perturbación (s)

Paso 4: Comparación

$\Delta E = c (s_{new}) - c (s)$

si $\Delta E > 0$ entonces

$s = s_{new}$

sino

si aleatorio $[0,1] < \exp (\Delta E / temp)$ entonces

$s = s_{new}$

fin si

fin si

fin desde

temp = $\alpha * temp$

fin mientras

Paso 5: Solución final s

5. Estructura de la solución

Un elemento importante para la aplicación del algoritmo de recocido simulado es la estructura de la solución. La estructura de la solución debe poder representar a cualquier solución del problema y a la vez debe ser adecuada para poder aplicar eficazmente la función perturbación. Como el PDBAH es la combinación de un problema de ruteo y un problema de asignación, la estructura de la solución está formada por dos partes. La figura 3 presenta un ejemplo de la estructura de la solución definida para el PDBAH. La primera parte, conjunto ruta, define las rutas que realizan los vehículos. Cada elemento del conjunto ruta es la ruta que realiza un vehículo (K), en un periodo (T) y en una ruta (R). El primer valor indica el número

total de PD a visitar, el resto los PD a visitar y su orden indica el orden en que serán visitados. No se incluye al CDL (nodo 0) porque toda ruta inicia y termina en el CDL. La segunda parte es la matriz asignación que establece la cantidad de ayuda humanitaria que se entrega en cada PD. Cada fila de la matriz asignación esta asociada a una ruta y cada columna representa a un PD, el valor de la matriz indica la cantidad de ayuda humanitaria a entregar en el PD.

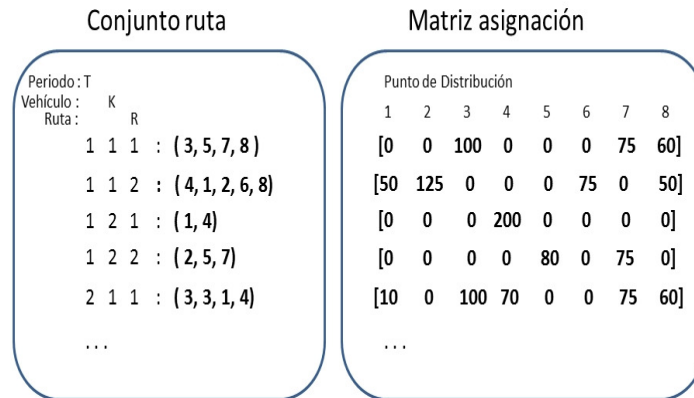


Fig. 3. Estructura de la solución para el PDBAH.

La figura 4 presenta esquemáticamente el algoritmo de recocido simulado desarrollado para el PDBAH. El algoritmo inicia con la inicialización de variables y la definición de una solución inicial factible que es generada aleatoriamente. El proceso de generación de la solución inicial tiene dos etapas en la primera etapa se generan rutas factibles para cada vehículo y en la segunda etapa se asignan cantidades a los PD visitados. Posteriormente inicia el ciclo externo y el ciclo interno. Este trabajo ha desarrollado una compleja función perturbación de la solución actual y obtener una nueva solución. La función esta formada de tres tipos de cambios: intercambio de cantidades entre vehículos, cambio en la ruta de un vehículo y cambio de cantidades al interior de una ruta. La función perturbación inicia cambiando la solución actual bien por un intercambio de cantidades entre vehículos o por un cambio en la ruta de un vehículo. Posteriormente se aplica el cambio de cantidades al interior de una ruta a diferentes rutas. Finalmente se obtiene una nueva solución. El cambio a aplicar se determina aleatoriamente, los pasos son: (1) generar un número aleatorio entre cero y uno según una distribución uniforme, (2) comparar el número aleatorio con el valor del parámetro ϵ para aplicar el cambio en la ruta de un vehículo y (3) si el número aleatorio es inferior a ϵ se aplica el cambio en la ruta de un vehículo sino se aplica el intercambio de cantidades entre vehículos. Siguiendo las etapas del algoritmo de recocido simulado la nueva solución es comparada con la solución actual y puede que la solución nueva se convierta en la solución actual. El algoritmo termina cuando la temperatura alcanza el criterio de paro.

La solución inicial comienza con una solución vacía. Para definir el conjunto rutas se recorre todas las rutas posibles. A cada ruta se añaden PD de manera aleatoria hasta que alcanza el tamaño de 5 PD o sobrepasa el tiempo de trabajo máximo del

vehículo. Previamente a concluir que se sobrepasa el tiempo de trabajo máximo, se analizan todas las posibles rutas. Si todas las rutas sobrepasan el tiempo de trabajo máximo se elimina el último PD añadido. Una vez definido el conjunto de rutas se define la matriz asignación. El primer paso es asignar una cantidad fija pequeña a cada PD visitado. El segundo paso recorre cada ruta, si está visita al menos un PD, se determina la cantidad a rellenar que es el mínimo entre la capacidad de vehículo restante y la ayuda humanitaria disponible. La ayuda humanitaria disponible se actualiza cada vez que una parte es asignada a un PD. La cantidad a rellenar se añade al primer PD visitado por la ruta. En el tercer paso, cada ruta pasa por una fase de mejora al aplicar el cambio de cantidades al interior de la ruta.

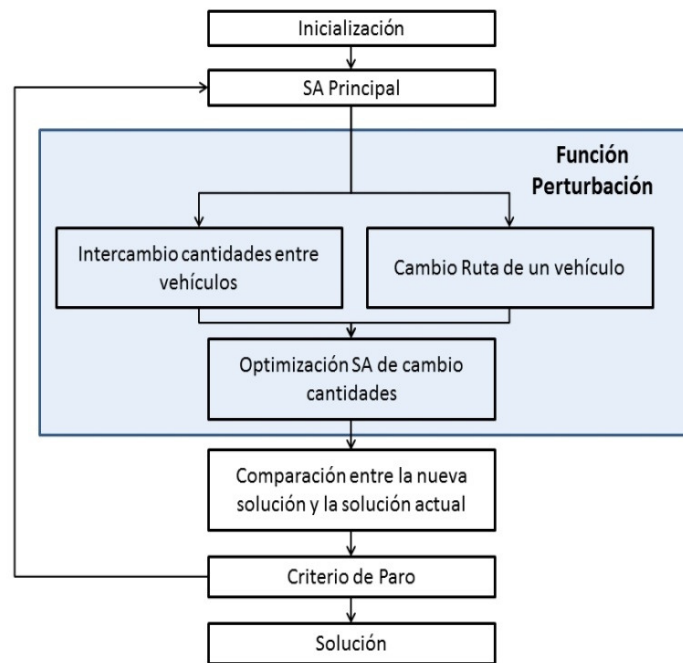


Fig. 4. Esquema del algoritmo de recocido simulado para el PDAH.

En el cambio de cantidades entre vehículos una cantidad de ayuda humanitaria que iba a ser entregada por un vehículo pasa a ser entregada por otro vehículo. Primero se selecciona un periodo y después dos vehículos-ruta de manera aleatoria según una distribución uniforme, la carga del segundo vehículo-ruta debe ser inferior a la capacidad de carga del vehículo. Luego se intercambian una cantidad, definida aleatoriamente según una distribución uniforme. La cantidad a intercambiar es reducida de un PD del primer vehículo-ruta para ser asignada a un PD del segundo vehículo-ruta. Las rutas modificadas pasan por una fase de mejora al aplicar el cambio de cantidades al interior de la ruta.

El cambio de la ruta de un vehículo modifica la ruta al incluir un nuevo PD. El primer paso es seleccionar una ruta de un vehículo de un periodo de manera aleatoria según una distribución uniforme. Posteriormente se añade un PD nuevo y diferente de los PD visitados por la ruta de manera aleatoria como último nodo visitado. Si la ruta tiene seis PD se elimina al primer PD visitado por la ruta. En la DAH las rutas son cortas y en general no tienen más de tres PD, ya que la capacidad de los vehículos respecto de las necesidades de los PD es relativamente pequeña. Se verifica que la ruta con el nuevo PD sea factible, es decir que se cumple el tiempo de trabajo máximo. Para lo cual se validan todas las rutas posibles hasta que se encuentra una que sea factible, sino se puede formar una ruta factible se eliminan PD según el orden de visita original de la ruta, hasta que se forma una ruta factible o se queda vacía. Una vez modificada la ruta, el segundo paso es la reasignación de las cantidades de ayuda humanitaria. De manera similar al cambio de cantidades entre vehículos, la nueva ruta pasa por una fase mejora al aplicarle el cambio de cantidades al interior.

El cambio de cantidades al interior de una ruta únicamente modifica las cantidades entregadas de una ruta y modifica únicamente la matriz de asignación. El proceso recorre cada ruta de la solución que tiene más de un 1 PD. Para cada ruta se genera un número aleatorio mediante una distribución uniforme entre cero y uno. Si es menor al parámetro de probabilidad de selección la ruta es seleccionada para aplicarle el cambio. Si la ruta es seleccionada, se le aplica un algoritmo de recocido simulado en el cual la función perturbación intercambia cantidades entre los PD visitados en la ruta. Para realizar el intercambio se selecciona dos PD y una cantidad a intercambiar de manera aleatoria. Se realiza el cambio generando una nueva solución.

6. Algoritmo mateheurístico

Raidl y Puchinger [15] indican que sólo en los últimos años muchos investigadores reconocen las ventajas y potencialidades de la construcción de métodos híbridos entre los métodos exactos y (meta)heurísticos. Es importante observar que los métodos exactos y los métodos (meta)heurísticos tienen sus propias ventajas y desventajas, y pueden ser vistos como complementarios, en gran medida. Parece natural combinar las ideas de ambos métodos. En la comunidad científica se ha establecido el término mateheurística para referirse a los algoritmos híbridos entre métodos exactos y métodos (meta)heurísticos. Puchinger y Raidl [14] presentan una clasificación general de los algoritmos mateheurísticos agrupados en dos categorías:

- Combinaciones colaborativas: En un ambiente de colaboración los algoritmos intercambian información, pero son independientes. Los algoritmos exactos y (meta)heurísticos pueden ejecutarse secuencialmente, en paralelo o entrelazadamente.
- Combinaciones integradas: En los métodos integrados, un algoritmo es un componente subordinado de otro algoritmo.

En la categoría de combinaciones integradas, Caserta y Vob [4] y Raidl y Puchinger [15] identifican dos subcategorías (i) el algoritmo (meta)heurístico es el maestro y controla las llamadas al algoritmo exacto y (ii) el algoritmo exacto es el maestro y llama al algoritmo (meta)heurístico.

```

Paso 1: Inicialización
s = solución inicial
temp =  $T_{ini}$  (Temperatura inicial)
 $\alpha$  = factor de reducción
iter = número de iteraciones
 $T_{fin}$  = Temperatura final (criterio de paro)
Paso 2: Búsqueda
mientras Temp >  $T_{fin}$ 
  desde x=1 a iter
    Paso 3: Nueva solución
    (i)  $sp_{rutas}$ , establecer la solución parcial ruta ( $s = sp_{rutas} + sp_{asig}$ )
         $sp_{asig}$ , establecer la solución parcial asignación
    (ii)  $sp_{new\ rutas} = Perturbar ( sp_{rutas} )$ 
    (iii)  $P(sp_{new\ rutas})$ , establecer Problema de asignación de ayuda humanitaria
    (iv)  $sp_{new\ asig} = Optimizar P( sp_{new\ rutas} )$ 
    (v)  $s_{new} = sp_{new\ rutas} + sp_{new\ asig}$ 
    (vi) Depurar ( $s_{new}$ )
    Paso 4: Comparación
 $\Delta E = c ( s_{new} ) - c ( s )$ 
    si  $\Delta E > 0$  entonces
      s =  $s_{new}$ 
    sino
      si aleatorio [0,1] <  $\exp ( \Delta E / temp )$  entonces
        s =  $s_{new}$ 
      fin si
    fin si
  fin desde
  temp =  $\alpha * temp$ 
fin mientras
Paso 5: Solución final s

```

* el operador + une dos soluciones parciales para formar una solución completa

Fig. 5. Algoritmo mateheurístico para el PDAH.

El algoritmo mateheurístico desarrollado es un algoritmo de recocido simulado con una función perturbación más sofisticada que incluye la resolución de un subproblema de programación entera. El pseudo código del algoritmo mateheurístico desarrollado se presenta en la figura 5. El algoritmo inicia con la definición de los parámetros del algoritmo. Se define la solución inicial de manera aleatoria, de forma igual a la presentada para el algoritmo de recocido simulado. El paso 2 inicia el algoritmo de recocido simulado con el ciclo exterior e interior. El paso 3 genera una nueva solución a partir de la solución actual. Las etapas son: (i) Se separa la solución actual (s) en dos partes la solución parcial rutas (sp_{rutas}) y la solución parcial asignación (sp_{asig}). (ii) La solución parcial rutas (sp_{rutas}) se modifica al aplicarle varias veces el cambio de la ruta de un vehículo, definiendo la nueva solución parcial rutas ($sp_{newrutas}$). (iii) Se define un problema de asignación de ayuda humanitaria asociado a la nueva solución parcial rutas ($P(sp_{newrutas})$). (iv) El problema se resuelve óptimamente al utilizar programación

entera generando la nueva solución parcial asignación ($s_{p_{newasig}}$). (v) En esta etapa se unen las dos soluciones parciales nueva generando una nueva solución (s_{new}). (vi) Se depura la nueva solución (s_{new}) eliminando de las rutas a los PD que no reciben ayuda humanitaria. Continuando con el algoritmo principal de recocido simulado, se realiza la comparación entre la nueva solución (s_{new}) y la solución actual (s). De la comparación puede que se produzca un cambio en la solución actual (s). El algoritmo termina cuando la temperatura alcanza el criterio de paro.

7. Implementación y resultados experimentales

El presente trabajo define un conjunto de instancias de manera aleatoria. Se definieron dos tipos de instancias: pequeñas y grandes. Las instancias pequeñas están formadas por 10 PD y las instancias grandes están formadas por 20 PD. La localización de los PD fue generada de manera aleatoria según una distribución uniforme en un plano con un rango de 100 X 100 unidades de distancia. En todas las instancias la localización del CDL es en el centro del plano con coordenadas [50,50]. Los tiempos de recorrido entre los nodos se establece según la distancia euclídea redondeada hacia abajo en unidades de tiempo. La demanda de cada PD en cada periodo fue generada a partir de una distribución uniforme ($U[1000,6000]$). Una vez establecida la demanda de cada PD para cada día, se establece la oferta disponible para cada periodo también utilizando una distribución aleatoria uniforme. Para la oferta disponible se establece entre el 80 y el 95 % de la demanda de ese periodo. Se recuerda que la capacidad de transporte está definida por cuatro elementos: cantidad de vehículos, capacidad de carga, tiempo de trabajo máximo (TTM) y el número de rutas. Se definen instancias con 2 vehículos y capacidad de carga de 8000 y 6000, para el tipo pequeño y con 2 vehículos y capacidad de carga de 16000 y 12000, para el tipo grande. El TTM se presenta con dos opciones por cada opción de cantidad de vehículos y capacidad de carga. El número de rutas máximo que puede realizar un vehículo esta limitado a 2 rutas por periodo. En consecuencia, se identifican cuatro grupos definidos por el número de PD y el número de vehículos. Para cada tipo de problema se generaron 6 problemas base en sus datos de localización, tiempos de recorrido, demanda y oferta. El número total de instancias es de 24.

Los pesos de la función objetivo se mantuvieron constantes en todas las instancias. Sus valores son: P3: 100, P2: 40 y P1: 10. La función convexa lineal por segmentos utilizada para el objetivo de balance en la demanda no satisfecha se presenta en la figura 6 y está compuesta de 10 segmentos.

utilizado es la técnica exacta de programación entera con el algoritmo brach-and-bound y se utiliza el software Cplex versión 11.0. Para terminar la ejecución se determina un tiempo de proceso límite de 7200 segundos. El segundo método utilizado es el algoritmo de recocido simulado desarrollado, el algoritmo es implementado en Python 2.7.2. Los valores de los parámetros utilizados en el algoritmo son: $Temp_{ini}$: 10000, $Temp_{fin}$: 1000, factor α : 0.95 e iter: 5 y ϵ : 0.75. Los parámetros del cambio de cantidades al interior de una ruta son: $Temp_{ini}$: 10000, $Temp_{fin}$: 6000, factor α : 0.95 y iter: 5. El tercer método utilizado es un algoritmo metaheurístico desarrollado. Este algoritmo es implementado en Python 2.7.2 y el problema de programación entera es resuelto por el software Gurobi 4.6. Los valores

de los parámetros del algoritmo son: $Temp_{ini}$: 10000, $Temp_{fin}$: 1000, factor α : 0.95 e iter: 20. Los valores de los parámetros fueron determinados en base al conocimiento de los algoritmos desarrollados y de su desempeño en experimentos iniciales del PDBAH. Todos los experimentos se realizaron en una computadora con un procesador AMD Athlon 64 de 1.80 GHz con 2.00 Gb de memoria RAM. Todos los tiempos de proceso en este trabajo están reportados en segundos.

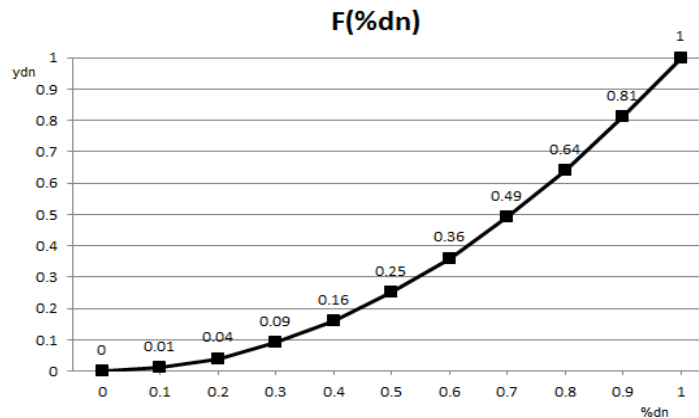


Fig. 6. Función convexa lineal por segmentos de la demanda no satisfecha.

Para resolver las instancias definidas se utilizan tres métodos. El primer método El algoritmo recocido simulado y el algoritmo mateheurístico utilizan números aleatorios, por lo cual estos métodos realizan 5 replicas de resolución de cada instancia. Los resultados presentados reportan los promedios del valor del objetivo y del tiempo de proceso o el tiempo de CPU consumido de las 5 replicas. Así mismo, el valor del objetivo es comparado con la cota inferior que se obtiene al resolver las instancias mediante el método exacto. La figura 7 presenta los resultados obtenidos para cada escenario y método. Al final de la figura 7 se presenta los promedios totales de los resultados obtenidos. La columna GAP representa la diferencia en porcentaje entre la solución obtenida y la cota inferior respecto de la cota inferior. En la figura 5 se observa en la columna del valor del objetivo del método exacto que en ocasiones aparece como dato 'No sol', esto indica que el algoritmo no encontró ninguna solución entera que respete las restricciones, al terminar el tiempo de proceso limite.

Esto se demuestra observando que en todos los escenarios obtiene la solución promedio con mejor valor del objetivo y el mejor GAP. El promedio entre todas las instancias del algoritmo mateheurístico obtiene un 0.35% del GAP. El promedio del GAP que obtiene el algoritmo de recocido simulado (SA) es el 7%. El método exacto obtiene un promedio del 208% del GAP, en 1 instancia del tipo grande no logra obtener una solución entera factible. También se puede comparar los resultados obtenidos por el algoritmo de recocido simulado y por el algoritmo mateheurístico en la figura 8.

Distribución balanceada de ayuda humanitaria resuelto con un algoritmo metaheurístico

Instancia	Problema	PD	TTM	Exacto			SA			MATEHEURISTICO			
				Valor Función Obj	Cota Inferior	GAP	Tiempo Proceso	Valor Función Obj	GAP	Tiempo Proceso	Valor Función Obj	GAP	Tiempo Proceso
1	1	10	300	5,121,408	1,916,588	167%	7,200	2,162,580	13%	634	1,918,003	0.07%	129
2	2	10	300	11,226,810	3,820,891	194%	7,200	4,171,668	9%	556	3,828,887	0.21%	121
3	3	10	300	5,981,182	3,032,790	97%	7,200	3,124,079	3%	537	3,036,039	0.11%	127
4	4	10	300	11,034,009	2,743,189	302%	7,200	3,849,502	40%	449	2,876,119	4.85%	116
5	5	10	300	14,372,314	3,197,490	349%	7,200	3,267,655	2%	606	3,200,647	0.10%	127
6	6	10	300	3,985,805	3,013,989	32%	7,200	3,353,307	11%	494	3,022,965	0.30%	128
7	1	10	400	1,916,593	1,916,588	0%	7,200	1,920,908	0%	713	1,916,588	0.00%	149
8	2	10	400	4,797,129	3,820,891	26%	7,200	3,836,965	0%	602	3,820,891	0.00%	135
9	3	10	400	3,032,796	3,032,790	0%	7,200	3,052,907	1%	617	3,032,790	0.00%	140
10	4	10	400	2,760,085	2,743,189	1%	7,200	2,767,675	1%	634	2,743,189	0.00%	139
11	5	10	400	3,197,495	3,197,490	0%	7,200	3,198,478	0%	673	3,197,490	0.00%	147
12	6	10	400	5,467,352	3,013,989	81%	7,200	3,025,753	0%	627	3,013,989	0.00%	139
13	1	20	400	32,096,500	5,275,081	508%	7,200	5,815,175	10%	1,156	5,350,924	1.44%	240
14	2	20	400	No Sol	8,529,787		7,200	8,752,039	3%	1,045	8,529,965	0.00%	249
15	3	20	400	23,507,754	3,482,580	575%	7,200	4,035,957	16%	1,265	3,486,212	0.10%	245
16	4	20	400	13,747,862	6,158,087	123%	7,200	6,412,637	4%	1,055	6,168,677	0.17%	254
17	5	20	400	31,779,275	4,217,081	654%	7,200	4,667,486	11%	1,094	4,253,217	0.86%	252
18	6	20	400	37,145,737	9,390,788	296%	7,200	9,485,239	1%	1,120	9,390,790	0.00%	269
19	1	20	600	25,400,108	5,275,081	382%	7,200	6,061,551	15%	1,187	5,275,082	0.00%	259
20	2	20	600	10,145,539	8,529,787	19%	7,200	8,626,229	1%	1,049	8,529,787	0.00%	261
21	3	20	600	26,715,600	3,482,580	667%	7,200	4,090,414	17%	1,286	3,482,581	0.00%	252
22	4	20	600	6,248,632	6,158,087	1%	7,200	6,313,219	3%	1,078	6,160,792	0.04%	260
23	5	20	600	4,289,479	4,217,081	2%	7,200	4,742,448	12%	1,105	4,222,815	0.14%	256
24	6	20	600	37,391,000	9,390,788	298%	7,200	9,472,025	1%	1,064	9,390,788	0.00%	264
Promedio:						208%	7,200		7%	860		0.35%	194

Fig. 7. Resultados de la experimentación.

Los resultados muestran que el algoritmo mateheurístico tiene el mejor desempeño. Respecto del tiempo de proceso, se observa que el algoritmo mateheurístico requiere un consumo de tiempo de proceso inferior a los otros métodos, 194 segundos en promedio frente a los 860 segundos del algoritmo de recocido simulado. El método exacto siempre termina al alcanzar el límite de tiempo. En la figura 5 se observa que al duplicar el número de puntos de distribución se duplica el tiempo de proceso, en el caso del recocido simulado y el algoritmo mateheurístico.

El TTM es un elemento que dificulta de manera inversa la resolución del problema. En la figura 5 en la columna del GAP del método exacto se observa que obtiene mejores soluciones (con el mismo tiempo de proceso) cuando el TTM es alto en comparación con un TTM bajo. Esto es debido a que un TTM bajo es más restrictivo que un TTM alto. Sin embargo esta tendencia no se observa tan claramente en los otros métodos, salvo en las instancias con 10 PD resueltas con el algoritmo de recocido simulado. Se observa que el valor del TTM no afecta al tiempo de proceso.

La figura 5 se observa que en algunas instancias el método exacto obtiene buenas soluciones con GAP inferiores a 5% en 6 instancias y que los peores resultados se presentan cuando el TTM es bajo. El algoritmo de recocido simulado obtiene buenos resultados en 14 instancias con GAP menores al 5%.

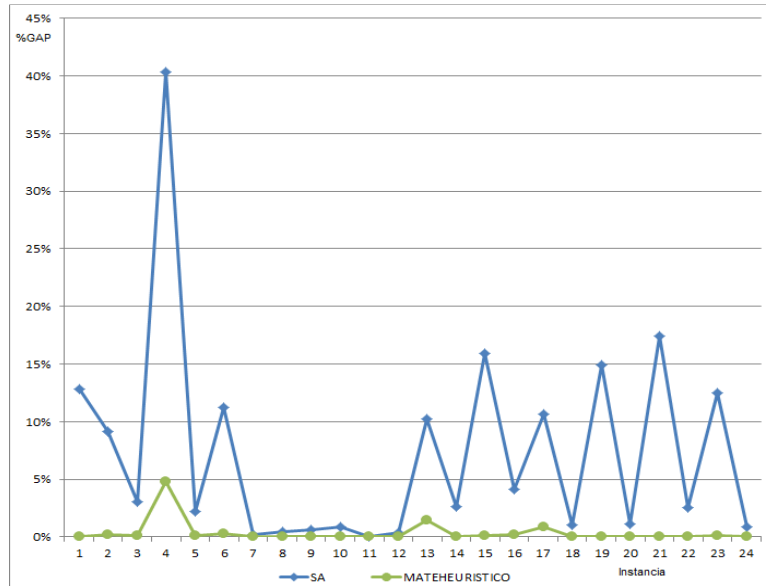


Fig. 8. Gráfica de los resultados en el GAP del recocido simulado y del mateheurístico.

8. Conclusiones

En el presente trabajo se define un modelo para el PDABH. También se ha desarrollado un algoritmo de recocido simulado y un algoritmo mateheurístico.

Una dificultad del PDABH es definir su modelo. En este trabajo desarrolla un modelo a partir de la formulación de dos índices del problema de ruteo. Primero se modifica al incluir múltiples periodos por la limitación de recursos. La limitación en la ayuda humanitaria también obliga a satisfacer parcialmente la demanda de los puntos de distribución. La tercera modificación es incluir la satisfacción tardía de la demanda, inventarios en los puntos de distribución y la realización de múltiples rutas. La última modificación es incluir de manera explícita la realización de una distribución balanceada a través de una función convexa lineal por segmentos para evitar grandes diferencias en el servicio recibido por los puntos de distribución.

El algoritmo mateheurístico desarrollado es comparado con el método exacto branch-and-bound y el algoritmo de recocido simulado. El algoritmo mateheurístico desarrollado descompone la solución en una parte de ruteo y otra parte de asignación. La parte de ruteo es la más difícil y por ello es tratada en el algoritmo principal basado en el algoritmo de recocido simulado y la parte de asignación más sencilla es tratada con programación entera. Para la comparación del desempeño se definen 96 instancias que son resueltas por los tres métodos. En base a los resultados obtenidos se concluye que el algoritmo mateheurístico desarrollado se desempeña mejor que las otras técnicas en todas las instancias en términos del valor del objetivo y del tiempo de proceso utilizado.

Referencias

1. Altay N., Green III W.G., OR/MS research in disaster operations management. *European Journal of Operational Research*, 175: 475-493, 2006.
2. Balcik B., Beamon B.M., Smilowitz K., Last Mile Distribution in Humanitarian Relief. *Journal of Intelligent Transportation Systems*, 12(2): 51-63, 2008.
3. Barbarosoglu G., Ozdamar L., Cevik A., An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations. *European Journal of Operational Research*, 140: 118-133, 2002.
4. Caserta M., Vob S., Metaheuristics: intelligent problem solving. En Maniezzo V., Stutzle T., Vob S. *Metaheuristics*, New York (NY), Springer, 2009, pp. 1-38.
5. De Angelis V., Mecoli M., Nikoi C., Storchi G., Multiperiod integrated routing and scheduling of World Food Programme cargo planes in Angola. *Computers Operations Research*, 34: 1601-1615, 2007.
6. Dowsland K., Diaz B., Diseño de heurísticas y fundamentos del recocido simulado. *Revista Iberoamericana de Inteligencia Artificial*, 20: 34-52, 2001.
7. Ergun O., Karakus G., Keskinocak P., Swann J., Villarreal M., Operations Research to improve disaster supply chain management. *Wiley Encyclopedia of Operations Research and Management Science*, Hoboken (NJ), John Wiley & Sons, 2011.
8. Huang M., Balcik B., Smilowitz K., Models for relief routing: Equity, efficiency and efficacy. *Transportation Research Part E*, 48(1): 2-18, 2012.
9. Kirkpatrick S., Gelatt C.D., Vecchi M.P., Optimization by Simulated Annealing. *Science*, 220(4598): 671-680, 1983.
10. Kovacs G., Spens K.M., Humanitarian logistics in disaster relief operations. *International Journal of Physical Distribution & Logistics Management*, 37(2): 99-114, 2007.
11. Lin Y., Batta R., Rogerson P.A., Blatt A., Flanigan M. A logistics model for delivery of critical items in a disaster relief operation: heuristic approaches. Disponible en: <http://www.acsu.buffalo.edu/~batta/tre.pdf>.
12. Nolz P.C., Doerner K.F., Gutjahr W.J., Hartl R.F., A bi-objective metaheuristic for disaster relief operation planning. En Coello Coello C., Dhaenens C., Jourdan L., *Advances in multi-objective nature inspired computing*, Berlin, Springer Berlin / Heidelberg, 2010, pp. 167-187.
13. Perez N., Holguin-Veras J., Mitchell J.E., Sharkey T.C., Integrated vehicle routing problem with explicit consideration of social costs in humanitarian logistics. En proceso de publicación, 2010. Disponible en: <http://transp.rpi.edu/~HUM-LOG/Doc/IRP.pdf>.
14. Puchinger J., Raidl G.R., Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. *Lecture Notes in Computer Science*, 3562: 41-53, 2010.
15. Raidl G.R., Puchinger J., Combining (Integer) Linear Programming Techniques and Metaheuristics for Combinatorial Optimization. *Studies in Computational Intelligence*, 114: 31-62, 2008.
16. Sheu J., An emergency logistics distribution approach for quick response to urgent relief demand in disasters, *Transportation Research Part E*, 43: 687-709, 2007.
17. Toth, P. Vigo, D. The vehicle routing problem. *Society for industrial and applied mathematics*, Philadelphia (PA), 2002.
18. Van Wassenhove L.N., Humanitarian aid logistics: supply chain management in high gear. *Journal of the Operational Research Society*, 57(5): 475-489, 2006.
19. Yi W., Ozdamar L., A dynamic logistics coordination model for evacuation and support in disaster response activities. *European Journal of Operational Research*, 179: 1177-1193, 2007.

Robótica e interfaces inteligentes

Control sub-óptimo con desempeño robusto para regulación global de robots manipuladores

Mario O. Ordaz-Oliver, Omar J. Santos-Sánchez
y Omar A. Domínguez-Ramírez

Universidad Autónoma del Estado de Hidalgo,
Grupo de Electrónica y Control, CITIS-ICBI, Mineral de la Reforma, Hidalgo, México
x2_kyo@hotmail.com, omarj@uaeh.edu.mx, omar@uaeh.edu.mx

Resumen En este trabajo se experimenta con un controlador sub-óptimo no lineal basado en el enfoque de programación dinámica y la teoría de estabilidad de Lyapunov. La propuesta reportada en este documento conserva la sencillez del Regulador Cuadrático Lineal (LQR, de las siglas en inglés "Linear Quadratic Regulator"), pero considerando el caso no lineal. También se encuentran las condiciones suficientes que definen la estabilidad asintótica del sistema en lazo cerrado y se penaliza el índice de desempeño del controlador para su implementación en tareas de regulación global. Se presenta el resultado experimental de la contribución en una tarea de regulación basada en seguimiento aplicada en la interfaz háptica PHANToM 1.0

Palabras clave: sub-óptimo; no lineal; interfaz háptica; estabilidad asintótica; función de Bellman.

1. Introducción

Cuando se considera el problema del control óptimo para sistemas no lineales en el enfoque de programación dinámica, generalmente debe resolverse la ecuación de Hamilton Jacobi Bellman. Sin embargo, la obtención de la solución de la ecuación de Hamilton Jacobi Bellman no es una tarea fácil. Ahora, cuando la programación dinámica se aplica al control de robots, el problema se complica debido a las singularidades y no linealidades en los robots. Además, la dificultad aumenta cuando la función Bellman o una aproximación de esta tiene que ser propuesta con el fin de optimizar el desempeño dinámico del robot. En el área de control óptimo de robots, se han presentado principalmente dos tipos de resultados: planificación de trayectorias con la optimización de la ley de control y el problema de estabilización óptima. En [1] se desarrolla una ley de control óptima para el seguimiento de la trayectoria de robots móviles, empleando una linealización del modelo con el propósito de obtener un control óptimo mediante la resolución de una ecuación de Riccati, que considera un índice de desempeño cuadrático de horizonte infinito. Por otro lado, considerando que la dinámica

es desconocida, [6] emplea un regulador cuadrático lineal (LQR) para obtener un control robusto de un robot tipo SCARA de dos articulaciones. Utilizando la ecuación de Hamilton-Jacobi-Bellman, los autores obtienen el control óptimo para un sistema robótico mediante la solución de una ecuación algebraica de Riccati. La optimización cuadrática se combina con un modelo computacional cerebral aritmético y redes neuronales. [10] utiliza una transformación para solucionar el problema de seguimiento de trayectoria en tiempo óptimo como un problema de control óptimo convexo con una sola entrada, los resultados se validan únicamente por simulaciones sobre un robot manipulador industrial KUKA 361 de seis grados de libertad. [3] propone dos diferentes problemas de planificación de trayectorias para robots manipuladores. En el primer problema, el efector final del robot es obligado a moverse a través de una trayectoria prescrita en el espacio de trabajo. Para el segundo caso, la trayectoria del efector final se determina tomando en cuenta la presencia de obstáculos con restricciones holonómicas. Ambos problemas se resolvieron aplicando el enfoque de cálculo variacional y considerando un problema de control óptimo de horizonte finito en un robot manipulador SCARA. En [2] se resuelve el problema de redundancia óptima global para robots manipuladores con el principio del máximo de Pontryagin. En [4] se realiza un algoritmo para la solución numérica del problema de control óptimo, con una técnica de programación cuadrática recursiva. En general, en la literatura examinada, sólo se presentan resultados a nivel de simulación, sin abordar su validación experimental. Por lo tanto, la principal contribución de este trabajo es poner a prueba un controlador no lineal sub-óptimo en lazo cerrado basado en el enfoque de programación dinámica para estabilizar y regular un manipulador robótico, penalizando el índice de desempeño para obtener un comportamiento apropiado para robots manipuladores en regulación global. Esta ley de control fue probada previamente para controlar un vehículo autónomo no tripulado (UAV) [9], dando como resultado un buen rendimiento en tiempo real. Se debe considerar que este tipo de sistemas son subactuados, y en el presente trabajo, se ha probado la ley de control no lineal para un sistema robótico completamente actuado. Se presentan los resultados experimentales para validar la viabilidad de este controlador. Además, teniendo en cuenta algunas propiedades de los robots manipuladores y el análisis de estabilidad de Lyapunov, se verifica que el sistema robótico en lazo cerrado es asintóticamente estable. Con este enfoque, las penalizaciones del controlador no lineal probado no son muy difíciles de encontrar. Esto es una ventaja cuando la implementación de un controlador se realiza en línea. El esquema propuesto se valida con los resultados experimentales mediante el uso de un dispositivo háptico PHANTOM 1.0, en una tarea de regulación punto a punto. Además, utilizando una penalización adecuada del estado y de la ley de control en el índice de desempeño para la síntesis del controlador no lineal, los resultados experimentales obtenidos indican que el algoritmo de control sub-óptimo tiene un desempeño eficaz.

El documento está organizado de la siguiente manera: En la sección 2, se presentan algunos resultados preliminares: síntesis de la ley de control y prueba

de la estabilidad del sistema en lazo cerrado. En la sección 3 se describe la plataforma PHANTOM 1,0 y a partir de ello se definen las condiciones para la evaluación de la ley de control sub-óptimo. La Sección 4, se emplea para dar a conocer los resultados experimentales de la implementación. Finalmente, en la sección 5 se definen las conclusiones y observaciones finales.

2. Síntesis de la estrategia de control no lineal

En esta sección se sintetiza la estrategia de control que se aplicará para esta contribución. Se desarrolla una ley de control no lineal sub-óptima para un dispositivo háptico cuya dinámica resulta ser una sistema no lineal afín. Como es conocido en la literatura, un sistema no lineal libre de perturbaciones se representa como:

$$\dot{\bar{x}} = f_0(\bar{x}) + f_1(\bar{x})u, \quad (1)$$

en el cual $\bar{x} \in \mathbb{R}^{n \times 1}$ es el estado del sistema y $u \in \mathbb{R}^{1 \times p}$ es la entrada de control, con funciones continuas $f_0(\bar{x})$ y $f_1(\bar{x})$ en espacios apropiados. Suponga que el sistema (1) tiene un punto de equilibrio en cero y que las funciones $f_0(\bar{x})$, $f_1(\bar{x})$ satisfacen las siguientes condiciones:

$$\|f_0(\bar{x})\| \leq \alpha_0 \|x\|, \alpha_1 \leq \|f_1(\bar{x})\|, \alpha_0, \alpha_1 > 0. \quad (2)$$

Ahora considere la interfaz háptica Phantom 1.0 en la forma estándar para un robot manipulador:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau. \quad (3)$$

con $H(q) \in \mathbb{R}^{3 \times 3}$ matriz de inercias, $C(q, \dot{q}) \in \mathbb{R}^{3 \times 3}$ matriz de fuerzas centrífugas y de Coriolis, $G(q) \in \mathbb{R}^{3 \times 1}$ vector de fuerzas debidas a efectos gravitacionales, $\tau \in \mathbb{R}^{3 \times 1}$ vector de pares inerciales, q , \dot{q} y $\ddot{q} \in \mathbb{R}^{3 \times 1}$ vectores de posiciones, velocidades y aceleraciones generalizadas respectivamente, con la forma:

$$H(q) = \begin{bmatrix} h_{11}(q_2, q_3) & 0 & 0 \\ 0 & h_{22} & h_{23}(q_2, q_3) \\ 0 & h_{32}(q_2, q_3) & h_{33} \end{bmatrix},$$

$$C(q, \dot{q}) = \begin{bmatrix} c_{11}(q_2, q_3, \dot{q}_2, \dot{q}_3) & c_{12}(q_2, q_3, \dot{q}_1) & c_{13}(q_2, q_3, \dot{q}_1) \\ c_{21}(q_2, q_3, \dot{q}_1) & 0 & c_{23}(q_2, q_3, \dot{q}_3) \\ c_{31}(q_2, q_3, \dot{q}_1) & c_{32}(q_2, q_3, \dot{q}_3) & 0 \end{bmatrix},$$

$$G(q) = \begin{bmatrix} 0 \\ g_2(q_2) \\ g_3(q_3) \end{bmatrix}, \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

y todos sus parámetros conocidos [8].

Para obtener la forma estándar de un sistema no lineal de las ecuaciones dinámicas del robot, se tiene que:

$$\ddot{q} = \begin{bmatrix} \dot{q} \\ -H^{-1}(q)\{C(q, \dot{q})\dot{q} + G(q)\} \end{bmatrix} + \begin{bmatrix} 0_{n \times n} \\ H^{-1}(q)\tau \end{bmatrix}. \quad (4)$$

Observe que la matriz $H(\cdot)$ siempre posee inversa debido a que el sistema es completamente actuado. Defina las siguientes variables de estado: $q_1 = x_1$, $q_2 = x_2$, $q_3 = x_3$, $\dot{q}_1 = x_4$, $\dot{q}_2 = x_5$, $\dot{q}_3 = x_6$, y definiendo $\bar{x}_1 = [x_1 \ x_2 \ x_3]^\top$, $\bar{x}_2 = [x_4 \ x_5 \ x_6]^\top$, $\tau = u$, se reescribe (4) como

$$\dot{\bar{x}} = \begin{bmatrix} \bar{x}_2 \\ -H^{-1}(\bar{x}_1) \{C(\bar{x}) \bar{x}_2 + G(\bar{x}_1)\} \end{bmatrix} + \begin{bmatrix} 0_{n \times n} \\ H^{-1}(\bar{x}_1) u \end{bmatrix}, \quad (5)$$

donde

$$H(\bar{x}_1) = \begin{bmatrix} h_{11}(x_2, x_3) & 0 & 0 \\ 0 & h_{22} & h_{23}(x_2, x_3) \\ 0 & h_{32}(x_2, x_3) & h_{33} \end{bmatrix},$$

$$C(\bar{x}) = \begin{bmatrix} c_{11}(x_2, x_3, x_5, x_6) & c_{12}(x_2, x_3, x_4) & c_{13}(x_2, x_3, x_4) \\ c_{21}(x_2, x_3, x_4) & 0 & c_{23}(x_2, x_3, x_6) \\ c_{31}(x_2, x_3, x_4) & c_{32}(x_2, x_3, x_6) & 0 \end{bmatrix},$$

$$G(\bar{x}_1) = \begin{bmatrix} 0 \\ g_2(x_2) \\ g_3(x_3) \end{bmatrix}, u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix},$$

teniendo por consiguiente que

$$f_0(\bar{x}) = \begin{bmatrix} \bar{x}_2 \\ -H^{-1}(\bar{x}_1) \{C(\bar{x}) \bar{x}_2 + G(\bar{x}_1)\} \end{bmatrix}, \quad (6)$$

$$f_1(\bar{x}) = \begin{bmatrix} 0_{3 \times 3} \\ H^{-1}(\bar{x}_1) \end{bmatrix}. \quad (7)$$

Para garantizar $\|f_0(\bar{x})\| \leq \alpha_0 \|\bar{x}\|$ reescribiendo el sistema (3) con la dinámica de la interfaz háptica en forma de (1), en donde $f_0(\bar{x})$ se define como:

$$f_0(\bar{x}) = \bar{f}_0(\bar{x}) + \bar{g}_0(\bar{x}), \quad (8)$$

esta separación se hace dado que f_0 no satisface la condición de Lipschitz por la acción de los efectos gravitatorios. Tomando esta separación, (1) se redefine como:

$$\dot{\bar{x}} = \bar{f}_0(\bar{x}) + \bar{g}_0(\bar{x}) + f_1(\bar{x}) u, \quad (9)$$

en donde

$$\bar{f}_0(\bar{x}) = \begin{bmatrix} \bar{x}_2 \\ -H^{-1}(\bar{x}_1) C(\bar{x}) \bar{x}_2 \end{bmatrix}, \bar{g}_0(\bar{x}) = \begin{bmatrix} 0_{3 \times 1} \\ -H^{-1}(\bar{x}_1) G(\bar{x}_1) \end{bmatrix},$$

consecuentemente (1) se expresa como:

$$\dot{\bar{x}} = \begin{bmatrix} \bar{x}_2 \\ -H^{-1}(\bar{x}_1) C(\bar{x}) \bar{x}_2 \end{bmatrix} + \begin{bmatrix} 0_{3 \times 1} \\ -H^{-1}(\bar{x}_1) G(\bar{x}_1) \end{bmatrix} + \begin{bmatrix} 0_{3 \times 3} \\ H^{-1}(\bar{x}_1) \end{bmatrix} u. \quad (10)$$

Motivados por el hecho de que el sistema (10) puede ser linealizado alrededor de un punto de equilibrio x^+ , su linealización se expresa como:

$$\dot{\bar{x}} = A\bar{x} + Bu, \quad (11)$$

en donde:

$$\left. \frac{\partial f_0(\bar{x})}{\partial \bar{x}} \right|_{x=x^+, u=u^+} = A, \quad \left. \frac{\partial f_1(\bar{x})}{\partial u} \right|_{x=x^+, u=u^+} = B. \quad (12)$$

Con el par (A, B) controlable y $\bar{x} = x - x^+$. Considerando ahora el índice de desempeño dado por

$$J_L = \int_0^{\infty} (\bar{x}^\top Q \bar{x} + u^\top R u) dt, \quad (13)$$

donde $Q \geq 0$, $Q = Q^\top \in \mathbb{R}^{6 \times 6}$ y $R \geq 0$, $R = R^\top \in \mathbb{R}^{3 \times 3}$.

Es bien conocido que el sistema lineal (11) puede ser estabilizado de manera óptima resolviendo el problema LQR, esto es encontrando una función $V_B(\bar{x})$ tal que satisfaga la siguiente ecuación:

$$\left. \frac{dV_B(\bar{x})}{dt} \right|_{(11, u=u^*)} + \bar{x}^\top Q \bar{x} + (u^*)^\top R u^* = 0 \quad (14)$$

con $u^* = -R^{-1}B^\top P \bar{x}$. Entonces la función de Bellman que satisface la ecuación (14) se define por la forma cuadrática $V_B(\bar{x}) = \bar{x}^\top P \bar{x}$, donde P es la solución a la ecuación algebraica de Riccati.

Si se considera la ecuación (14), notese que el sistema (1) es localmente estable en el sentido de Lyapunov, en lazo cerrado con u^* . Como (11) es una linealización de (1), este modelo y control lineal u^* solo es válido para una región ε , tal que $\|\bar{x}\| < \varepsilon$ para algún $\varepsilon > 0$. Se desea una ley de control no lineal, tal que garantice la estabilidad en lazo cerrado del sistema no lineal (1). Entonces, para $\|\bar{x}\| > \varepsilon$ proponiendo la función de Bellman $V_B(\bar{x})$ como una aproximación de la ecuación de Bellman para el sistema (1) de la siguiente manera:

$$\left. \frac{dV_B(\bar{x})}{dt} \right|_{(1)} + \bar{x}^\top Q \bar{x} + u^\top R u$$

Derivando ahora la función de $V_B(\bar{x})$ a lo largo de las trayectorias del sistema (1) se obtiene:

$$D(\bar{x}, u) = 2f_0(\bar{x})P\bar{x} + 2\bar{x}^\top P f_1(\bar{x})u + \bar{x}^\top Q \bar{x} + u^\top R u, \quad (15)$$

note que $D(\bar{x}, u)$ es convexa respecto a u , debido a que $R > 0$. Consecuentemente, la ecuación (15) tiene un mínimo, que está dado por:

$$\frac{\partial D(\bar{x}, u)}{\partial u} = 2Ru + 2f_1(\bar{x})^\top P \bar{x} = 0,$$

entonces, se tiene que

$$\tilde{u} = -R^{-1}f_1^\top(\bar{x})P\bar{x}, \quad (16)$$

observe que si $\bar{x} \rightarrow 0$, entonces $f_1^\top(\bar{x}) \rightarrow B^\top$ y $\tilde{u} \rightarrow u^*$ en la región $\|\bar{x}\| < \varepsilon$.

Ahora, con el fin de obtener las condiciones suficientes para concluir que el sistema (1) es asintóticamente estable en lazo cerrado con el controlador (16), observe que

$$\begin{aligned} \left. \frac{dV(\bar{x})}{dt} \right|_{(1, u=\bar{u})} &= 2f_0(\bar{x})P\bar{x} + 2\bar{x}^\top P f_1(\bar{x})\tilde{u}, \\ &= 2f_0(\bar{x})P\bar{x} - 2\bar{x}^\top P f_1(\bar{x})R^{-1}f_1^\top(\bar{x})P\bar{x}, \end{aligned}$$

ya que $f_0(\bar{x})$ tiene la propiedad de Lipschitz, implica que

$$\left. \frac{dV(\bar{x})}{dt} \right|_{(1, u=\bar{u})} \leq 2\alpha_0\bar{x}^\top P\bar{x} - 2\bar{x}^\top P f_1(\bar{x})R^{-1}f_1^\top(\bar{x})P\bar{x},$$

entonces el sistema (1) es asintóticamente estable en lazo cerrado con el control (16) sí

$$\alpha_0 P - \alpha_1 \lambda_{\min}(R^{-1})P^2 < 0 \quad (17)$$

Ahora se puede establecer el siguiente resultado.

Proposición 1: Considere el sistema (1) con las funciones $f_0(\cdot)$ y $f_1(\cdot)$ que satisfacen las propiedades dadas en (2). Suponga que el sistema (1) puede ser linealizado alrededor de cierto punto de equilibrio x^+ y que el par (A, B) del sistema linealizado (11) es controlable. Entonces la ley de control sub-óptima u dada por la ecuación (16) estabiliza el sistema (1) sí la desigualdad (17) se cumple.

3. Control del dispositivo háptico PHANToM 1.0 y condiciones de evaluación

En esta sección se describe la arquitectura del hardware de la interfaz háptica PHANToM 1.0 y se obtienen las matrices de penalización Q y R del índice de desempeño dado por (13), las cuales son útiles para hallar la matriz P de ganancias del control no lineal (16). Nuestro objetivo es estabilizar el dispositivo bajo una acción de regulación, utilizando la ley de control no lineal propuesta anteriormente, por medio su interfaz de potencia y una computadora personal. La interfaz de potencia se compone por tres Procesadores Digitales de Señales (DSP, de sus siglas en inglés) con hardware en arquitectura cerrada, que se comunican con la computadora por medio del protocolo IEEE 1284EP en puerto paralelo. Los actuadores del dispositivo háptico cuentan con decodificadores ópticos HEDM-5500-B02 y tienen una resolución mayor a 1024 pulsos por revolución, con una resolución en posición nominal de 0.03 N/mm, una fricción de 0.004 N, fuerza máxima de 8.5 N, rigidez de 3.4 N/mm y una inercia (masa aparente en el efector final) de 75 gr. El ordenador cuenta con un procesador intel core 2Duo a 2.66 GHz que trabaja en ambiente PC. La programación tanto de la estrategia de control como la dinámica del sistema se hacen en lenguaje Visual C++ con el juego de herramientas H3D, para la visualización gráfica del desempeño se emplea una programación fuera de línea con MATLAB.

PHANToM 1.0 es un dispositivo háptico de tres grados de libertad, que cuenta con articulaciones de revolución y por tanto puede ser definido como un mecanismo de eslabones articulados, el cual tiene propósitos distintos a los de un robot manipulador, no obstante, la técnica de modelado matemático es la misma. Su modelo dinámico se emplea para el diseño de estrategias de control y esta constituido por las ecuaciones de movimiento que relacionan las fuerzas debidas a efectos inerciales, Coriolis, centrípetas, gravitatorias y de fricción, siendo esta última despreciable por la calidad en la transmisión de movimiento articular. El modelo dinámico reducido de este dispositivo es obtenido por medio de la formulación de Euler-Lagrange, al que es aplicado el cambio de variable $\bar{x} = x - x^+$ definido en la sección anterior, donde $u = [u_1 \ u_2 \ u_3]^\top$ corresponden a las entradas de control para cada uno de los actuadores del sistema. Los puntos de equilibrio se definen por $x^+ = [0 \ 0 \ 0 \ k_1 \ k_2 \ k_3]^\top$ que indican una tarea de regulación punto a punto a la posición articular $\theta_1 = k_1$, $\theta_2 = k_2$ y $\theta_3 = k_3$, en radianes y una velocidad 0 rad/seg. Suponiendo que (8) se satisface, $\bar{f}_0(\bar{x})$ cumple con la condición (2), puesto que los efectos debidos a la fuerza de gravedad sobre el dispositivo PHANToM 1.0 siempre se anulan en las componentes u_1 , u_2 y u_3 al agregar su compensación. Bajo este supuesto, $\bar{f}_0(\bar{x})$ satisface la condición (2) con $\alpha_0 = 8.660254037844387 \times 10^2$ y $f_1(\bar{x})$ también cumple la condición (2) con $\alpha_1 = 1.037690003471722 \times 10^4$. Inicialmente se utiliza la ley de control sub-óptima no lineal sintetizada en la sección 2, usando la notación estándar para el sistema no lineal dada en (1) y con el controlador óptimo de la ecuación (16), se estabilizará el sistema (1, 5). Proponiendo una linealización del sistema alrededor del punto de equilibrio $x^+ = [0 \ 0 \ 0 \ 0.5 \ 0.5 \ 0.5]^\top$, y con condiciones iniciales $x_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^\top$, se observa que el modelo linealizado del sistema (1) es controlable. En consecuencia, se calcula un controlador alrededor del punto de equilibrio x^+ , empleando el enfoque óptimo del LQR, entonces la función de Bellman se define como

$$V(x) = x^\top P x, P = P^\top > 0, \quad (18)$$

y P satisface la ecuación algebraica de Riccati. Para el índice de desempeño (13), se eligen las siguientes matrices Q y R :

$$Q = \begin{cases} q_{1,1} = 254.1756563965411, \\ q_{2,2} = 254.1756563965167, \\ q_{3,3} = 264.4382885838342, \\ q_{4,4} = 254.9973344114833, \\ q_{5,5} = 254.9973344114585, \\ q_{6,6} = 256.0342099346721, \\ q_{ij} = 0, \ i \neq j, \end{cases} \quad R = \begin{cases} r_{ij} = 6.153846153846154 \ i = j, \\ r_{ij} = 0 \ i \neq j. \end{cases}$$

Por lo tanto, el sistema linealizado alrededor del punto de equilibrio x^+ de la forma dada por (11) se define por las siguientes matrices

$$A = \begin{cases} a_{1,4} = 1, a_{2,5} = 1, a_{3,6} = 1 \\ a_{5,2} = 9.5839, a_{5,3} = -10.5219 \\ a_{6,2} = 37.8075, a_{6,3} = 386.5271 \\ 0 \text{ para los elementos restantes} \end{cases} \quad B = \begin{cases} b_{4,1} = 2535, b_{5,2} = 2549, b_{6,3} = 10377 \\ 0 \text{ para los elementos restantes} \end{cases}$$

La condición de estabilidad (2) se satisface porque $\alpha_0 P - \alpha_1 \lambda_{\min}(R^{-1}) P^2$ es definido negativo.

Observación 1: Se introducen algunas tendencias conservativas para la selección del par (Q, R) (una selección particular de estas matrices da como resultado una matriz P específica) que satisface la condición (17). Esto es una consecuencia natural, puesto que solo se proponen condiciones suficientes, de acuerdo a esto, se penaliza la convergencia del estado mediante Q y la robustez del control mediante R en (13) para obtener un desempeño del sistema aceptable en tareas de regulación. Por lo que las matrices Q y R escritas arriba, son el resultado de penalizar el estado y el control, hasta obtener un comportamiento deseable.

4. Resultados experimentales

En esta sección se presentan los resultados experimentales de la implementación de la ley de control no lineal sub-óptima, los cuales permiten validar el rendimiento y la estabilización del dispositivo háptico PHANTOM 1.0 en una tarea de regulación. En este experimento el dispositivo háptico se lleva a $x_1 = 0.5$ rad $x_2 = 0.5$ rad y $x_3 = 0.5$ rad, que corresponden a θ_1, θ_2 y θ_3 respectivamente, esto en un tiempo de 10 seg por medio de una regulación basada en seguimiento, que es obtenida a partir de un generador de tiempo base (TBG, de las siglas en inglés "Time Base Generator"), planteado para aplicaciones y esquemas de guiado háptico en [7]. Defina la función polinomial $\xi(t)$ para la posición articular deseada como se muestra en (19) y calcule su derivada para velocidad articular deseada.

$$\xi(t) = a_3 \frac{(t-t_0)^3}{(t_b-t_0)^3} - a_4 \frac{(t-t_0)^4}{(t_b-t_0)^4} + a_5 \frac{(t-t_0)^5}{(t_b-t_0)^5}, \quad (19)$$

donde los coeficientes $a_3 = 10$, $a_4 = 15$ y $a_5 = 6$, y $t_b = 10$ segundos, correspondiente a la duración del experimento. La Figura (1), ilustra el seguimiento articular, con alto desempeño evidente en la gráfica, para finalmente estabilizar al dispositivo en la consigna de movimiento propuesto, con error de baja magnitud al inicio del experimento.

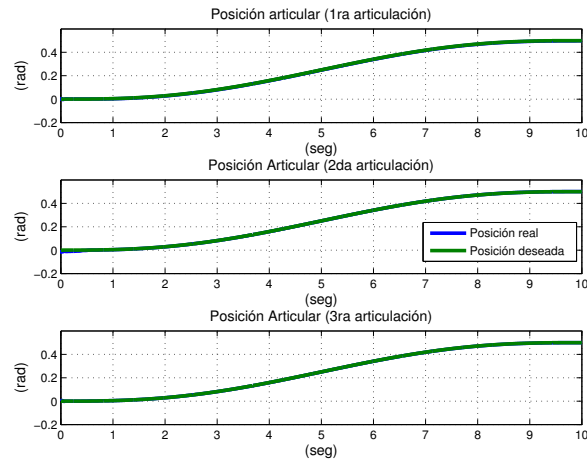


Figura 1. Posición articular y referencias deseadas del dispositivo háptico.

La Figura (2) representa la señal de error de posición articular, con errores máximos de centésimas de radián para la segunda articulación, dado el esfuerzo al que es sujeto el dispositivo al buscar la estabilización en la consigna articular. El error articular para la primera y tercera articulación, implica mayor desempeño, dado el menor esfuerzo en la ejecución de la tarea, adquiriendo magnitudes no mayores a 5 milésimas de radián.

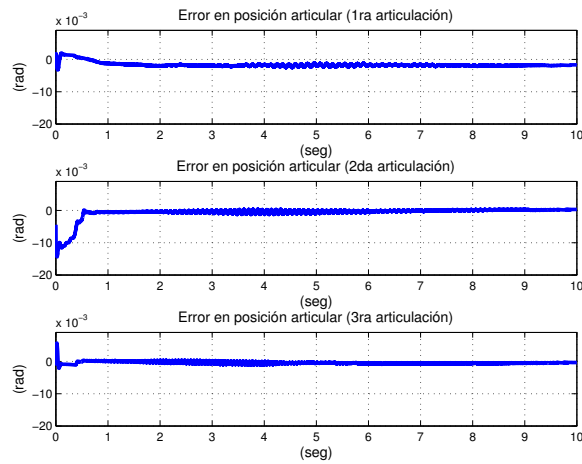


Figura 2. Señal de error en posiciones articulares del dispositivo háptico.

La contribución más relevante de la estrategia de control propuesta, se hace evidente en la Figura (3), ya que logra convergencia de posición articular con relativa baja energía. En el presente artículo no se dan a conocer estudios comparativos con otras técnicas de control, sin embargo es notable que la energía que emplea el dispositivo con otras estrategias de control robusto como en [5,7], es comparativamente mayor que nuestra propuesta.

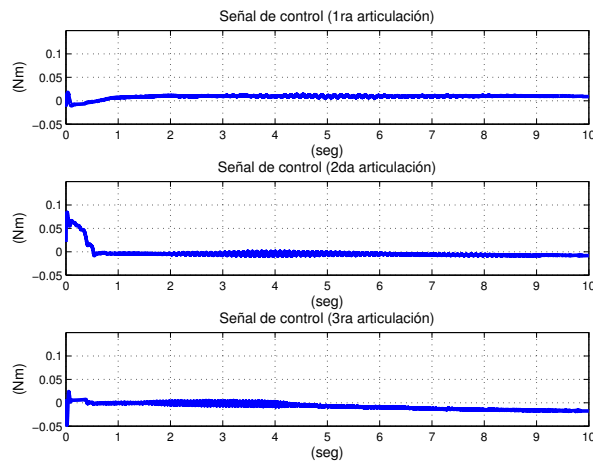


Figura 3. Señal de control aplicada al dispositivo háptica.

En la Fig. (4) se muestra el comportamiento de la energía total empleada en la tarea de regulación bajo la acción del controlador no lineal sub-óptimo, en ella puede observarse como la cantidad de energía requerida para la regulación es relativamente pequeña, y en particular es denominada energía potencial. El signo negativo de la energía total empleada en la tarea de regulación se debe a las características en la construcción del dispositivo, es decir, las condiciones de reposo (sin acción de control) del dispositivo PHANToM 1.0, se definen con el efector final en una posición vertical superior a su base y sin memoria mecánica en sus articulaciones, razón por la cual se incluye la compensación de la energía potencial del sistema en la ley de control.

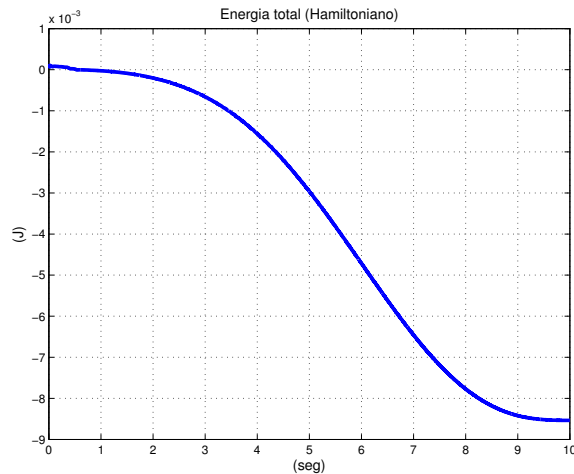


Figura 4. Energía total empleada en la tarea de regulación.

5. Conclusiones

El controlador sub-óptimo que se ha presentado en este artículo, para la estabilización del dispositivo háptico en tareas de regulación, ha tenido resultados experimentales satisfactorios. La contribución de este resultado ilustra la viabilidad de métodos sub-óptimos para problemas de minimización en el consumo de energía en el control de robots manipuladores (minimiza el esfuerzo del mecanismo de eslabones articulados e interfaces electromecánicas de potencia), y la obtención de un comportamiento robusto mediante la penalización de la velocidad de convergencia y del control en el índice de desempeño. Se demuestra como una función cuadrática es útil en la obtención de controladores no lineales sub-óptimos. Igualmente, la función cuadrática podría ser empleada como una función candidata a Lyapunov para robots manipuladores para el análisis de estabilidad en lazo cerrado. Se aprovecharon las propiedades del modelo del dispositivo háptico para sintetizar la estrategia de control no lineal propuesta. El algoritmo de control no lineal sub-óptimo se considera factible para su implementación en aplicaciones de robots manipuladores. Los experimentos presentados demuestran un rendimiento aceptable del dispositivo bajo la ley de control penalizada robustamente propuesta.

Agradecimientos. El primer autor agradece a **CONACYT** por la beca de maestría otorgada durante el periodo Agosto 2011- Junio 2013, con número de registro **290649**, al **CITIS-UAEH** y a su laboratorio de robótica avanzada e interfaces hápticas por el apoyo para el desarrollo del presente trabajo.

Referencias

1. Hemami, A. Mehrabi, M. G. and Cheng, R. M. H. Systhesis of an optimal control law for path tracking in mobile robots, *Automatica*, vol. 28(2), pp. 383-387, 1992.
2. Nakamura, Y., Optimal Redundancy Control of Robot Manipulators, *The International Journal of Robotics Research*, vol. 6(1), pp 32-42, 1987.
3. Gregoey, J., Olivares, A. and Stafetti, E. Energy-optimal trajectory planning for robot manipulators with holonomic constraints, *Systems & Control Letters*, vol. 61, pp. 279–291, 2012
4. Bock, HG., Plitt, KJ. A multiple shooting algorithm for direct solution of optimal control problems, *In proceedings of 9th IFAC World Congress*, Budapest, pp. 243–247, 1984.
5. Domínguez-Ramírez, O. A. & Parra-Vega, V. Active Haptic Interface with Purposes of Remote Training, *Proceedings of the 11th International Conference on Advanced Robotics*, ISBN 972-96889-9-0, vol. 3, pp. 1588-1593, University of Coimbra, Coimbra, Portugal, 2003.
6. Lin, F. and Brandt, R.D. An optimal control approach to robust control of robot manipulators, *IEEE Transaction on Robotics and Automation*, vol. 14(1), pp 69-77, 1998.
7. Parra-Vega, V., S. Arimoto, Y. H. Liu, G. Hirzinger, and P. Akella, Dynamic Sliding PID Control for Tracking of Robot Manipulators: Theory and Experiments, *IEEE Transaction on Robotics and Automation*, Vol. 19 (6), pp 967-976, December, 2003.
8. Domínguez-Ramírez, O. A. and Parra-Vega, V. Realistic Active Haptic Guided Exploration with Cartesian Control for Force-Position Tracking in Finite Time. *Journal of Applied Bionics and Biomechanics*, ISSN 1176-2322, ABBI, Vol 3 (4) pp 279-289, Cambridge, England, 2006.
9. Santos, Omar J., Romero, Hugo, Salazar, Sergio and Lozano, Rogelio. Real-Time Stabilization of a Quadrotor UAV by Nonlinear Suboptimal and Optimal Control Strategies, *Journal of Intelligent & Robotic Systems*, published on line in 2012, DOI 10.1007/s10846-012-9711-8, 2012.
10. Verscheure, D., Demeulenaere, B., Sweveres, J. De Schutter, J. and Diehl, M. Time optimal path tracking for robots: a convex optimization approach, *IEEE Trasaction on Automatic Control*, vol. 54(10), pp. 2318-2327, 2009.

Modelo teórico y generación de instancias de prueba para problemas RCCN (Reducing Collisions in Constrained Navigation)

Josué Oregel, Héctor J. Puga, Martín Carpio, Víctor Zamudio,
Manuel Ornelas y Luis Mancilla Espinoza

División de Estudios de Posgrado e Investigación,
Instituto Tecnológico de León,
León, Guanajuato, México
{josue-oregel,hector.puga}@itleon.edu.mx
<http://posgrado.itleon.edu.mx>

Resumen La presencia de colisiones o encuentros entre objetos móviles se considera un problema que genera un impacto negativo en sistemas de navegación reales. En este artículo se introducen los problemas de navegación a los que hemos llamado *Reducing Collisions in Constrained Navigation (RCCN)*, los cuales están relacionados con problemas que buscan satisfacer restricciones de enrutamiento o planificación en navegación como Problemas de Enrutamiento de Vehículos (VRP) y Vehículos Guiados Autónomos (AGV), donde RCCN por sus características es diferente de ambos. Se presenta un modelo que permite detectar colisiones o encuentros entre objetos móviles en sistemas de navegación con restricciones RCCN y provee una plataforma para la reducción de las colisiones o encuentros entre los móviles del sistema. También se presenta un generador de instancias de prueba con las características de los problemas RCCN.

Palabras clave: navegación, colisiones, instancias de prueba.

1. Introducción

En este trabajo se entiende un sistema de navegación como un sistema que está compuesto por un entorno de navegación, rutas para navegar, objetos móviles (o simplemente móviles) y una planificación para navegar por las rutas.

En sistemas inteligentes de navegación se pretende que un móvil esté dotado con cierto nivel de autonomía para reaccionar y tomar decisiones basándose en las condiciones del entorno, pero contrario a lo que sucede con los seres vivos, el desplazamiento de un ente artificial no se considera una tarea trivial dado que se enfrenta a problemas como percepción y modelado del entorno y planificación del desplazamiento por sus trayectorias evitando conflictos como colisiones, puntos muertos y congestiones, requiriendo ninguna intervención humana [1,2,3,4,5,6].

La principal aportación de este artículo es la construcción de estructuras esenciales para la futura resolución del problema de reducción del total de colisiones

o encuentros entre móviles mediante técnicas de optimización inteligente en sistemas de navegación con las restricciones de una clase específica de problemas de satisfacción de restricciones de navegación a la que hemos llamado *Reducing Collisions in Constrained Navigation (RCCN)*, que son problemas relacionados con Problemas de Enrutamiento de Vehículos (Vehicle Routing Problems, VRP) y Vehículos Guiados Automatizados (Automated Guided Vehicles, AGV), pero con características propias que los diferencian de estos [7,8,9,10,11,12,13,14,15,16,17].

Este documento está estructurado de la siguiente manera: en la Sección 2 se presentan las restricciones de un problema RCCN y el modelo propuesto, en la Sección 3 se presenta el generador de instancias de prueba además de una nomenclatura de instancias generadas utilizando el generador programado en Java. Finalmente, se presentan conclusiones y se menciona el trabajo desarrollado actualmente y su aplicación futura.

2. El modelo para problemas Reducing Collisions in Constrained Navigation (RCCN)

2.1. Restricciones para problemas RCCN

- El entorno de navegación es modelado como un mapa de rutas considerado como un grafo no dirigido que conecta sus nodos (o vértices) con aristas navegables y es generado a partir de una nube de puntos.
- Una ruta es una sucesión de aristas conectadas por nodos del grafo navegables por objetos móviles.
- Un nodo puede pertenecer a más de una ruta de navegación.
- Una arista pertenece a máximo una ruta y tiene definido únicamente un sentido de navegación.
- Una ruta es navegable únicamente por un móvil y este pertenece únicamente a una ruta, por lo tanto, existen tantas rutas como móviles en el sistema.
- Las rutas están predefinidas y son fijas para todos y cada uno de los móviles dentro del sistema de navegación.
- Para propósitos de estudio, los móviles son considerados como elementos puntuales, es decir, sin dimensión.
- La velocidad de recorrido de las rutas es la misma para todos los móviles, es constante y toma valores discretos.
- Un evento de colisión ocurre si hay un encuentro entre móviles en el mismo nodo al mismo tiempo dentro del mapa de ruta, es decir, no está permitido alojar más de un móvil en un nodo a la vez excepto en nodos de inicio o fin de ruta, similar a lo que ocurre en terminales reales.
- Es posible asignar tiempos de espera ($te \mid te \in \mathbb{N} \cup \{0\}$) para cada móvil en cada nodo de su ruta, y puede ser cambiado en intervalos discretos de tiempo ($\Delta te \in \mathbb{N}$), para evitar posibles colisiones en nodos comunes de su ruta con otras rutas.
- El inicio del recorrido de las rutas es simultáneo para todos los móviles, es decir, no están permitidos los tiempos de espera en nodos de inicio de ruta.

2.2. Modelo para detección y estrategia para eliminar colisiones

El modelo está estructurado en un procedimiento de 3 pasos:

Paso 1. Cálculo de tiempo de ocupación de nodos: Suponiendo que se cuenta con un entorno de navegación y además se cuenta con rutas para navegar en el entorno, entonces se cuenta con un grafo del sistema con n nodos y con este grafo se obtienen dos conjuntos de datos que son los insumos del modelo:

- Una “*Matriz de Conectividad Pesada*” (Mc) de tamaño $n \times n$, que es una matriz de adyacencia con pesos que representa al entorno de navegación y que se obtiene del grafo. Cada entrada de esta matriz contiene la información de conectividad y tiempo de recorrido $t_{(i,j)}$ ¹ entre los nodos i y j del entorno de navegación (Tabla 1).

Tabla 1. Información de $t_{(i,j)}$ contenida en las entradas de la matriz Mc .

Valor	Representación
-1	No hay una arista entre los nodos i y j (no hay conectividad).
0	Es el mismo nodo ($i = j$) (no hay conectividad).
$t_{(i,j)} \in \mathbb{N}$	Tiempo para recorrer la arista entre los nodos i y j (hay conectividad).

- Un conjunto de m rutas R_k , donde cada ruta es representada por una sucesión de s nodos $n_{(k,l)}$ del grafo (Ecuación (1)). El valor de s es independiente para cada ruta.

$$R_k = n_{(k,1)}, n_{(k,2)}, \dots, n_{(k,l)}, \dots, n_{(k,s)} \text{ donde } 1 \leq k \leq m, 1 \leq l \leq s, k, s \in \mathbb{N} . \quad (1)$$

A partir de la información anterior, se calcula el tiempo de ocupación de un nodo de las rutas y se registra en una *Matriz de Colisiones* M_{col} . En esta matriz cada entrada representa el tiempo que toma a un móvil llegar a un nodo de su ruta más el tiempo que permanece el móvil ocupando el nodo. Para generar la matriz M_{col} , se utilizan las siguientes tres matrices de tamaño $m \times n$: la *Matriz de Tiempo de Recorrido Original* (Mto), la *Matriz de Tiempo de Espera* (Mte) y la *Matriz de Tiempo de Retardo* (Mtr).

- *Matriz Mto:* Cada entrada $to_{(k,i)}$ de esta matriz contiene el tiempo de recorrido desde el nodo inicial hasta cada uno de los s nodos en cada ruta k (Ecuación (2)):

$$to_{(k,i)} = to_{(k,1)} + to_{(k,2)} + \dots + to_{(k,i-2)} + to_{(k,i-1)} . \quad (2)$$

¹ Aunque es común en lenguaje matemático el uso de la notación t_{ij} para hacer referencia a las entradas de una matriz, en este documento se ha utilizado una notación para subíndices usando paréntesis y coma $t_{(i,j)}$ con la intención de facilitar a los autores un manejo menos confuso de los mismos.

Los posibles valores contenidos en las entradas de la matriz Mto son los presentados en la Tabla 2:

Tabla 2. Posibles valores de $to_{(k,i)}$ en las entradas de la matriz Mto .

Valor	Representación
-1	El nodo i no pertenece a la ruta k .
0	Es el nodo de inicio de la ruta k ($to_{(k,i)} = 0, i = 1$).
$to_{(k,i)} \in \mathbb{N}$	Tiempo para recorrer la ruta k desde el inicio hasta el nodo i .

- **Matriz Mte :** Esta matriz contiene los valores de tiempo de espera $te_{(k,i)}$ asignados a cada uno de los s nodos de las k rutas.
- **Matriz Mtr :** En esta matriz cada entrada $tr_{(k,i)}$ contiene el tiempo de retardo en los s nodos de las k rutas, provocado por la acumulación del tiempo esperado por los móviles en nodos anteriores de la ruta k (Ecuación (3)):

$$tr_{(k,i)} = te_{(k,1)} + te_{(k,2)} + \dots + te_{(k,i-2)} + te_{(k,i-1)} . \quad (3)$$

El intervalo de tiempo total $tt_{(k,i)}$ que un móvil permanece ocupando un nodo i de su ruta k , es el comprendido entre el tiempo acumulado $ta_{(k,i)}$ (Ecuación (4)) y la suma del tiempo acumulado con el tiempo de espera asignado en el nodo i (Ecuación (5)).

$$ta_{(k,i)} = to_{(k,i)} + tr_{(k,i)} . \quad (4)$$

$$tt_{(k,i)} = [ta_{(k,i)}, ta_{(k,i)} + te_{(k,i)}] . \quad (5)$$

La matriz M_{col} almacena los intervalos de tiempo $tt_{(k,i)}$ y es el arreglo que permite al modelo realizar la detección de colisiones en un sistema de navegación con las características de los problemas RCCN.

Paso 2. Cálculo de eventos de colisión: Cuando un nodo i común a dos rutas k y l no es inicio o final de una de las rutas y existe una intersección en los intervalos de tiempo total de recorrido hasta el nodo, se identifica un evento de colisión por $c_{([k,l],i)}$ (Ecuación (6)). Para obtener el total de colisiones TCS en el sistema de navegación, se calcula la suma del total de eventos de colisión en cada nodo del sistema (Ecuación (7)). El Algoritmo 1 presenta el pseudocódigo para el cálculo de eventos de colisión.

$$c_{([k,l],i)} = \begin{cases} 1 & \text{si } tt_{(k,i)} \cap tt_{(l,i)} \neq \emptyset \\ 0 & \text{de otra forma} \end{cases}, \text{ donde } i \neq 1, i \neq s . \quad (6)$$

$$TCS = \sum_{i=1}^n \sum_{k=1}^{m-1} \sum_{l=k+1}^m c_{([k,l],i)} . \quad (7)$$

Algoritmo 1 Pseudocódigo del cálculo de eventos de colisión

Entrada: Una Matriz de Conectividad M_c **Entrada:** Un conjunto de m rutas de navegación

- 1: Calcular los intervalos de tiempo $tt_{(k,i)}$ de las entradas de la Matriz de Colisiones M_{col}
 - 2: Inicializar contador de colisiones $TSC = 0$
 - 3: **para** (cada nodo i del sistema de navegación; $1 \leq i \leq n$) **hacer**
 - 4: **para** (cada ruta k del sistema de navegación; $1 \leq k \leq m - 1$) **hacer**
 - 5: **para** (cada ruta l del sistema de navegación; $j + 1 \leq l \leq m$) **hacer**
 - 6: **si** ($tt_{(k,i)} \cap tt_{(l,i)} \neq \emptyset$) **entonces**
 - 7: Aumentar el contador de colisiones $TSC = TSC + 1$
 - 8: **fin si**
 - 9: **fin para**
 - 10: **fin para**
 - 11: **fin para**
 - 12: **devolver** Contador de colisiones TSC
-

Paso 3. Estrategia para eliminación de eventos de colisión: Para eliminar un evento de colisión entre objetos móviles en un nodo común a dos rutas diferentes, se selecciona una de las rutas y se establece un cambio en el tiempo de espera Δte del nodo anterior al nodo en conflicto (Ecuación (8)). Este cambio en el tiempo de espera produce el retardo en la llegada del móvil de la ruta elegida hacia el nodo común con evento de colisión, llegando después del móvil que se desplaza por la otra ruta, eliminando el evento de colisión.

$$te_{(k,i-1)} = te_{(k,i-1)} + \Delta te, \text{ donde } i \text{ es un nodo con evento de colisión .} \quad (8)$$

Es posible asignar en los nodos valores arbitrarios de tiempo de espera para reducir colisiones; sin embargo, como es deseable que en todo sistema de navegación se cumpla con el recorrido de las rutas en el menor tiempo posible, es recomendable asignar tiempos de espera cercanos a cero para afectar lo menos posible el tiempo de recorrido original.

3. Generador de instancias de prueba

Para probar el rendimiento de propuestas de solución a problemas RCCN es necesario contar con instancias de prueba adecuadas. En el estado del arte existen instancias de prueba para problemas VRP, AGV y más variantes, que se han utilizado en solución a problemas de navegación (e.g. generación de rutas de navegación óptimas, planificación de navegación sobre las rutas, etc.). Sin embargo, no se han encontrado instancias que pudieran usarse para probar el rendimiento de propuestas de solución a problemas RCCN conservando las características del modelo presentado, por lo que ha sido necesario generar un conjunto de instancias de prueba confiable y adecuado.

En esta sección se presenta un generador de instancias de prueba que cumple con las restricciones de problemas RCCN, la construcción se explica en dos pasos: la generación de rutas de navegación y la generación de tiempos de recorrido de las rutas de navegación.

3.1. Generación de rutas de navegación

El generador se basa en una nube de n puntos (o nodos) como entorno para la generación de rutas de navegación. Las rutas de navegación están definidas como una sucesión de nodos y cada ruta se compone de una cantidad independiente de nodos, lo que permite la aplicación de una entre dos opciones de configuración: la misma cantidad de nodos para cada ruta o cantidades diferentes de nodos para cada ruta.

Si se considera el conjunto de rutas generadas como parte de un sistema de navegación, entonces el número de *aristas totales* At requeridas para generar las rutas del sistema se calcula a partir de la cantidad s_k de nodos de cada ruta, (Ecuación 9):

$$At = \sum_{k=1}^m (s_k - 1) . \quad (9)$$

Por otro lado, la cantidad de *aristas útiles* Au para generar rutas a partir de un nube de puntos de n puntos es (Ecuación (10)):

$$Au = \frac{n(n-1)}{2} . \quad (10)$$

Si $Au \geq At$, teóricamente es posible con los n nodos del grafo completo generar el sistema de navegación con At aristas, de otra forma, el sistema no se podrá generar. El Algoritmo 2 presenta el pseudocódigo para la generación de rutas de navegación.

3.2. Generación de tiempos de recorrido de rutas

Para generar colisiones usando las rutas generadas, se ubican los nodos comunes y se asigna tiempo de recorrido a cada segmento de las rutas, de tal forma que la suma de tiempo de recorrido hasta los nodos comunes sea la misma para cada ruta involucrada. El algoritmo 3 presenta el pseudocódigo para la asignación de tiempos de recorrido a los segmentos de ruta del sistema de navegación.

3.3. Instancias de prueba generadas

La generación de instancias de prueba se llevó a cabo con un programa en Java [18] que entrega dos archivos para cada instancia: un archivo `.mcc` que contiene la Matriz de Conectividad Pesada Mc y un archivo `.rtc` que contiene m rutas de navegación. Estos dos archivos son las dos estructuras de datos que se usan como insumo en el modelo para problemas RCCN.

Algoritmo 2 Pseudocódigo para la generación de rutas de navegación

Entrada: n : cantidad de nodos del sistema de navegación**Entrada:** m : cantidad de rutas que se van a generar**Entrada:** s_k : cantidad de nodos de R_k

```

1: para cada  $R_k$  desde 1 hasta  $m$  hacer
2:   seleccionar aleatoriamente un nodo y asignarlo a  $R_k$  como nodo inicial ( $n_{(k,1)}$ )
3:   para cada  $i$  desde 2 hasta  $s_k$  hacer
4:     repetir
5:       seleccionar aleatoriamente un nodo ( $n_{(k,i)}$ )
6:       si  $n_{(k,i)}$  es diferente de  $n_{(k,i-1)}$  (nodo anterior en la ruta) entonces
7:         si aún no existe una arista entre  $n_{(k,i)}$  y  $n_{(k,i-1)}$  entonces
8:           asignar  $n_{(k,i)}$  a  $R_k$ 
9:         fin si
10:      fin si
11:   hasta que ha sido asignado un  $n_{(k,i)}$  a  $R_k$ 
12:   fin para
13: fin para
14: devolver  $m$  rutas de navegación.

```

Las instancias tienen la nomenclatura: **#N-#R-#C-ver#**, donde N hace referencia al número de nodos, R al número de rutas, C al número de colisiones del sistema y puesto que es posible generar diferentes mapas de rutas con la misma configuración de nodos, *ver* hace referencia a la versión de la instancia generada, comenzando con 1.

El valor de semilla utilizado en el método *Random* del programa Java ha sido almacenado en los archivos generados para brindar la posibilidad de reproducir cada instancia de prueba.

Algunos ejemplos de nomenclatura de instancias de prueba generadas se muestran en la Tabla 3.

4. Conclusiones

La presencia de conflictos, como colisiones, en sistemas de navegación se considera un problema debido al impacto negativo que los conflictos provocan en los sistemas reales. En sistemas de navegación donde la cantidad de colisiones entre móviles es grande puede ser necesario contar con herramientas de planificación que mejoren el desempeño del sistema.

En este trabajo se ha presentado una propuesta que puede permitir un mejor desempeño de un sistema de navegación. La propuesta contiene los siguientes dos elementos:

- Un modelo para problemas a los que se ha llamado *Collision Reduction in Constrained Navigation (RCCN)*, que permite detectar colisiones o encuentros entre objetos móviles y provee una plataforma para reducir estas utilizando técnicas de optimización inteligente.

Algoritmo 3 Pseudocódigo para la asignación de tiempo de recorrido de rutas

Entrada: m rutas de navegación

```
1: para cada  $R_k$  desde 1 hasta  $m$  hacer
2:   para cada  $n_{(k,i)}$  desde 1 hasta  $s_k$  hacer
3:     si  $n_{(k,i)}$  es un nodo común de  $R_k$  y otra ruta entonces
4:       aumentar el contador de cantidad de nodos comunes en  $R_k$ 
5:     fin si
6:   fin para
7: fin para
8:  $mnc =$  índice de la ruta con la mayor cantidad de nodos comunes
9: asignar un tiempo de recorrido igual a cero a  $n_{(mnc,1)}$ 
10: para cada  $n_{(mnc,i)}$  desde 2 hasta  $s_{mnc}$  hacer
11:   asignar aleatoriamente un tiempo de recorrido a  $n_{(mnc,i)}$ 
12: fin para
13: para cada  $R_k$  desde 1 hasta  $m$  excepto  $mnc$  hacer
14:   para cada  $n_{(k,i)}$  desde 1 hasta  $s_k$  hacer
15:     si  $n_{(k,i)}$  es un nodo común entre  $R_k$  y  $R_{mnc}$  entonces
16:       generar un tiempo de recorrido hasta  $n_{(k,i)}$  igual al de  $n_{(mnc,i)}$ 
17:     fin si
18:   fin para
19: fin para
20: si aún hay nodos comunes entonces
21:   para cada  $R_k$  desde 1 hasta  $m - 1$  excepto  $mnc$  hacer
22:     para cada  $R_l$  desde  $R_{k+1}$  hasta  $m$  excepto  $mnc$  hacer
23:       para cada  $n_{(k,i)}$  desde 1 hasta  $s_k$  hacer
24:         para cada  $n_{(l,i)}$  desde 1 hasta  $s_l$  hacer
25:           si  $n_{(k,i)} = n_{(l,i)}$  entonces
26:             generar un tiempo de recorrido hasta  $n_{(k,i)}$  igual al de  $n_{(l,i)}$ 
27:           fin si
28:         fin para
29:       fin para
30:     fin para
31:   fin para
32: fin si
33: si hay trayectos sin tiempos de recorrido entonces
34:   asignar tiempo de recorrido para los trayectos faltantes
35: fin si
36: devolver una Matriz de Conectividad Pesada  $Mc$ 
```

Tabla 3. Ejemplos de nomenclatura de instancias de prueba RCCN generadas con un programa en Java.

Nombre de Instancia	Semilla Java Random	Nodos por ruta
27N-5R-5C-ver1	-3770157802445094294	27
27N-5R-7C-ver1	-6075749904103534460	27
27N-5R-8C-ver1	4480157765928300129	27
27N-5R-11C-ver1	53830088977007794	27
27N-5R-12C-ver1	-4538243863454794414	27
100N60R-676C-ver1	1242326594992500141	Entre 60 y 90
150N-90R-941C-ver1	242298037008080666	Entre 80 y 105
200N-80R-1089C-ver1	-7332834085541032903	Entre 120 y 200
250N-200R-4879C-ver1	3618874879153313709	Entre 100 y 130
300N-120R-1739C-ver1	6519364232554299876	Entre 200 y 300
400N-300R-10541C-ver1	2333333709128786646	Entre 150 y 250
500N-300R-10293C-ver1	4932274200351035357	Entre 250 y 390
1000N-650R-75841C-ver1	-5907896972978185640	Entre 700 y 770
2000N-1000R-91390C-ver1	-9174493999035787771	Entre 1,000 y 1,500

- Un generador de instancias de prueba con las características de problemas RCCN, que son la base para comparar el rendimiento de técnicas de optimización inteligente aplicadas a problemas RCCN.

El modelo se basa en un procedimiento de 3 pasos: Calcula el tiempo que cada móvil permanece ocupando cada uno de los vértices de su ruta, considerando el intervalo entre el instante en que el móvil llega al nodo hasta el instante en que lo abandona. Calcula el número de eventos de colisión presentes en el sistema de navegación, registrando un evento de colisión cada vez que se detecta más de un móvil en un nodo común a las rutas analizadas en el intervalo de tiempo de ocupación del nodo por parte de cada móvil. Finalmente, provee una estrategia para el ajuste de tiempo de espera que permanecerán los móviles en nodos anteriores a los nodos en conflicto de su ruta para evitar eventos de colisión con otros móviles.

El generador construye instancias en dos pasos: la generación en forma aleatoria de rutas en sistemas de navegación de n nodos y la generación de colisiones en nodos comunes entre las rutas mediante la asignación de tiempo de recorrido de las rutas.

Es importante mencionar que la aplicación del modelo permitirá, en primera instancia, el uso de técnicas de optimización inteligente en sistemas de navegación teóricos, para luego ser implementado en sistemas de navegación reales, como transporte de suministros en sistemas flexibles de manufactura, técnicas *store and forward* en redes de computadoras o coordinación de actividades para robots asistentes en oficinas. Actualmente se está trabajando en heurísticas basadas en técnicas de optimización como *Particle Swarm Optimization (PSO)* y *Artificial Immune System (AIS)*.

Agradecimientos. Los autores agradecen el soporte del Consejo Nacional de Ciencia y Tecnología (CONACYT) y de la Dirección General de Educación Superior Tecnológica (DGEST) proyecto número 4572.12-P.

Referencias

1. Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S.: Principles of robot motion: theory, algorithms, and implementations. MIT press (2005)
2. Jianyang, Z., Wen-Jing, H.: Conflict-free routing of agvs on the mesh topology based on a discrete-time model. In: Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on. Volume 3., IEEE (2003) 3510–3516
3. Kruusmaa, M., Willemson, J.: Covering the path space: a casebase analysis for mobile robot path planning. Knowledge-Based Systems **16**(5) (2003) 235–242
4. LaValle, S.: Planning algorithms. Cambridge university press (2006)
5. Mirtich, B.: Efficient algorithms for two-phase collision detection. Practical motion planning in robotics: current approaches and future directions (1997) 203–223
6. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. 2 edn. Pearson Education, Inc. (2002)
7. Baldoni, R., Bonnet, F., Milani, A., Raynal, M.: Anonymous graph exploration without collision by mobile robots. Information Processing Letters **109**(2) (2008) 98–103
8. Chiew, K., Qin, S.: Scheduling and routing of amos in an intelligent transport system. IEEE Transactions on Intelligent Transportation Systems **10-3** (09 2009) 547–552
9. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Management science **6**(1) (1959) 80–91
10. Gendreau, M., Potvin, J., Bräumlaysy, O., Hasle, G., Løkketangen, A.: Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. The vehicle routing problem: Latest advances and new challenges (2008) 143–169
11. Ghasemzadeh, H., Behrangi, E., Abdollahi, A.M.: Conflict-free scheduling and routing of automated guided vehicles in mesh topologies. Robotics and Autonomous Systems **57**(6-7) (2009) 738–748
12. Gong, F., Wang, X.: Robot path-planning based on triangulation tracing. In: Intelligent Information Technology Application Workshops, 2008. IITAW'08. International Symposium on, IEEE (2008) 713–716
13. Jinghui, L., Jingbo, S., Yan, Z.: A fast approach to judging basic element intersection in polygons meshing. In: Digital Manufacturing and Automation (ICDMA), 2011 Second International Conference on, IEEE (2011) 20–23
14. Jose, A.A., Adarsh, S.B., Anju C., S.P.: A novel approach for scheduling and routing of the self guided vehicles in mesh topology using velocity control and alternate path techniques. In: 2011 International Conference on Process Automation, Control and Computing (PACC), IEEE (2011) 1–5
15. Peasgood, M., Clark, C., McPhee, J.: A complete and scalable strategy for coordinating multiple robots within roadmaps. Robotics, IEEE Transactions on **24**(2) (2008) 283–292
16. Peng, J., Akella, S.: Coordinating multiple double integrator robots on a roadmap: Convexity and global optimality. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, IEEE (2005) 2751–2758

17. Qiu, L., Hsu, W., Huang, S., Wang, H.: Scheduling and routing algorithms for agvs: a survey. *International Journal of Production Research* **40**(3) (2002) 745–760
18. Oracle, I.: Java se documentation at a glance. <http://www.oracle.com/technetwork/es/java/javase/documentation/index.html> (01 2013)

Captura de movimiento de brazo derecho mediante una plataforma de evaluación para tener un banco de datos: Estudio experimental en paciente hemipléjico

Abraham Briseño Cerón ^{1,2}, Omar A. Domínguez Ramírez ¹, Vicente Parra Vega ³
y Dora A. Rodríguez Vega ²

¹Centro de Investigación en Tecnologías de Información y Sistemas,
Universidad Autónoma del Estado de Hidalgo,
Universidad Politécnica de Sinaloa,
México

³Centro de Investigación y de Estudios Avanzados, Unidad Saltillo,
México

omar@uaeh.edu.mx, {abriseno,drodriguez}@upsin.edu.mx,
vparra@cinvestav.mx

Resumen El estudio de las capacidades motoras de las extremidades superiores mediante la ayuda de sistemas visuales puede ser complejo debido al espacio de trabajo que normalmente está abarrotado de oclusiones. En este trabajo, esta limitación se elude con la propuesta de una plataforma para estudiar el movimiento articular en pacientes hemipléjicos, que recurre a un sistema de cámaras que trabajan en un plano 3D y permite estudiar al miembro superior derecho a siete grados de libertad basado en el modelo cinemático de la extremidad y la interfaz háptica para calificar las habilidades motoras. Bajo el esquema de protocolos diseñados bajo condiciones controladas. El primero se basa en espacial visual con el modelo cinemático Denavit Hartenberg para cuantificar los movimientos espaciales en planos anatómicos. La segunda se ejecuta una terapia física basada en el protocolo Peabody para establecer efectos de la actividad visual y motora utilizando el dispositivo háptico Phantom Omni. Los resultados experimentales en voluntarios sanos y en pacientes hemipléjicos conducirán a evaluar los rangos de movimiento y los umbrales que indican un daño en el miembro superior derecho, así como las directrices que se sugieren para el uso de la robótica en el diagnóstico útil para la rehabilitación motora.

Palabras clave: captura de movimiento, modelo cinemático, miembro superior.

1. Introducción

En este trabajo se presentan dos estudios experimentales, que permiten analizar y evaluar los daños ocurridos por un accidente cerebrovascular presentándose parálisis de medio cuerpo. Para esta investigación se analizó la información obtenida del miembro superior derecho, mediante la planificación de actividades y a través de un sistema de captura de movimiento corporal. Aunque existen muchos estudios sobre el

análisis de movimiento del miembro superior, cada experimento puede estar basado en un protocolo médico diferente o estudios específicos de algún comportamiento.

La naturaleza del movimiento del brazo es tan diferente por cada humano debido a que es variable y compleja [1]. Sin embargo para realizar un estudio del miembro superior, se toma como base una actividad rutinaria o planificada, tal y como se realiza en la fase de rehabilitación, debido a que la repetición de tareas establecen un mecanismo de recuperación de movimiento. La mayoría de los estudios establecen una tarea para obtener modelos o patrones, tomados del análisis de una población, para esto establecen rutinas donde el usuario interactúa con objetos y a través de la captura de imágenes o video, estudian los movimientos para determinar por ejemplo ángulos de movimiento o características del padecimiento [2].

Para esta investigación se tomaron muestras de personas sanas y con hemiplejía, desde un modelo 3D y son almacenados en archivos .xls ó texto, lo que permite analizar el comportamiento de los daños ocasionados en el miembro superior derecho. El primer experimento consiste en el análisis del movimiento para identificar niveles de daño por hemiplejía, para realizarlo se usó un sistema de captura de movimiento [3] y la planificación de movimientos biomecánicos. En el segundo estudio se utilizó el protocolo clínico PeaBody, un guante de datos de cinco sensores [5], un phantom omni [4] y el sistema de captura, para observar el impacto al realizar las actividades del protocolo clínico a través del dispositivo robótico.

Este estudio es la base para modelar algoritmos de planificación de movimiento y fuerza empleados en sistemas robóticos como el cyberforce y cybergrasp, para la etapa de rehabilitación en miembro superior derecho.

2. Metodología

Para llevar a cabo la obtención de los datos con la captura de movimientos, se siguen los pasos que se muestran en la figura 1.

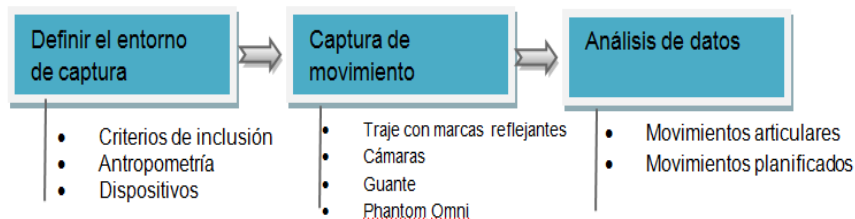


Figura 1. Metodología para la caracterización.

- Definir el entorno de captura: Primeramente se definen los criterios de inclusión, así mismo se planifica los movimientos que realizarán las personas y se define el entorno y dispositivos que participaran para que el muestreo sea homogéneo para personas sanas y con hemiplejía.
- Para la captura de movimiento se utilizó un traje con marcas reflejantes en puntos que permiten capturar los movimientos articulares, con el guante se mide la flexión y extensión de la mano en el agarre del gimbal del phantom

omni. Con este último dispositivo se realizan movimientos de acuerdo al protocolo Pea body, que con la lectura de datos se ve el impacto en los movimientos sobre el miembro superior derecho.

- c) Para analizar los datos se generaron archivos en Excel y Texto, a partir del modelo en 3D que proporciona el software de captura de movimiento, a través del código desarrollado con el SDK del fabricante del sistema de cámaras.

3. Desarrollo

Para estas muestras participaron 7 personas sanas (3 sexo femenino y 4 sexo masculino) y 3 pacientes del sexo femenino con hemiplejía, bajo las siguientes condiciones ambientales:

- Temperatura regulada a 25° centígrados.
- Humedad en rango de 60% al 80%.
- Iluminación a 1100 lúmenes.
- La edad de las personas sanas es de 20 a 30 años, estudiantes, no fuman, no toman y de complexión delgada.
- Personas con hemiplejía: Edad 40 a 50 años, amas de casa, sexo femenino, padecimiento de 1 a 6 meses, observaciones médicas similares de presenta secuelas de infarto cerebral afectando hemicuerpo derecho, presenta problemas de lenguaje, cuerpo derecho funcional en un 20%, el miembro superior derecho (MSD) a un 10% de contracturas en flexión, hombro derecho doloroso, el tobillo derecho inestable por lo que se usa ortesis, valoración recomendada cada 6 semanas.

Este trabajo de investigación está dividido en dos experimentos que a continuación se mencionan.

Tabla 1. Parámetros D-H para miembro superior.

Unión	β_i	Número	α_i	a_i	d_i	θ_i
Base	0	1 _(0,1)	0	a_0	d_0	0
Hombro	(-90 rotación media/ rotación lateral +90)	2 _(1,2)	-90	0	0	$\beta_1 + 90^\circ$
Hombro	(-180 abducción/ aducción +50)	3 _(2,3)	90	0	0	$\beta_2 + 90^\circ$
Hombro	(-180 flexión/ extensión +80)	4 _(3,4)	0	L1	0	$\beta_3 + 90^\circ$
Codo	(-10 flexión/ extensión +145)	5 _(4,5)	90	0	0	$\beta_4 + 90^\circ$
Codo	(-90 pronación/ supinación +90)	6 _(5,6)	90	0	L2	$\beta_5 + 90^\circ$
Muñeca	(-90 flexión/ extensión +70)	7 _(6,7)	90	0	0	$\beta_6 + 90^\circ$
Muñeca	(-15 abducción/ aducción +40)	8 _(7,8)	0	L3	0	β_7

3.1. Primer experimento

Para el análisis de movimiento de miembro superior derecho se planificaron los movimientos de estudio basándose en el modelo Denavit-Hartenberg D-H [6], para 7 grados de libertad (gdl) en miembro superior, como se observa en la figura 2.

En la Tabla 1, se muestran los parámetros D-H que se tomaron de referencia para este estudio [6].

El rango de movimiento para cada variable en el modelo, θ_i , depende del rango fisiológico de movimiento para la articulación anatómica correspondiente, β_i . Los parámetros a_i y d_i , son longitudes de los segmentos corporales, que son constantes para cada individuo [6].

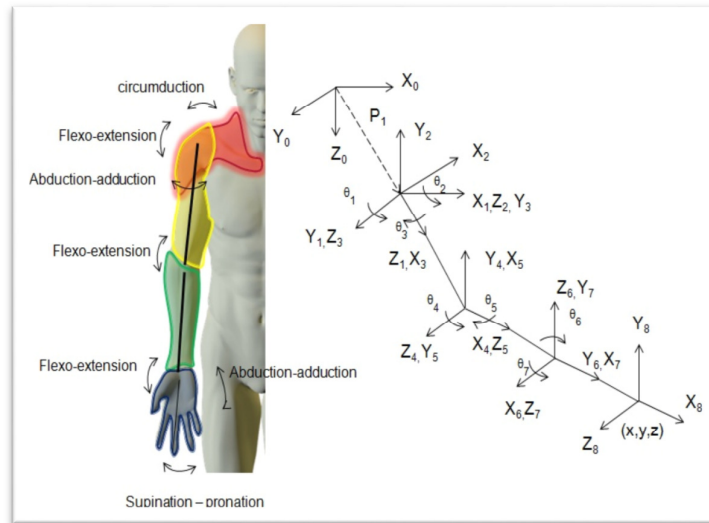


Figura 2. Modelado a 7 grados de libertad del miembro superior con notación D-H [6].

Los movimientos articulares para evaluar los daños ocasionados en el movimiento articular por hemiplejía sugeridos son:

- Circunducción: movimiento circular que combina la flexión, extensión, separación y aproximación, de tal manera que el extremo distal de la estructura efectúa un círculo.
- Flexión: movimiento que reduce el ángulo entre los huesos o partes del cuerpo.
- La extensión: movimiento que aumenta el ángulo entre los huesos de la extremidad en una articulación.
- Abducción: movimiento hacia afuera de la extremidad lejos del plano medio del cuerpo.
- Aducción es un movimiento que trae un miembro - brazo o una pierna -

más cerca del cuerpo, lo opuesto a la abducción.

- Supinación es la rotación del antebrazo, la palma hacia arriba.
- Pronación es una rotación del antebrazo la palma hacia abajo

La captura de movimiento, para este experimento se realizó como se observa en la figura 3 y mediante la selección entre las marcas se obtuvo información como longitud, la posición tridimensional de las marcas, entre otra, permitiendo calcular los ángulos articulares en cada movimiento planificado.

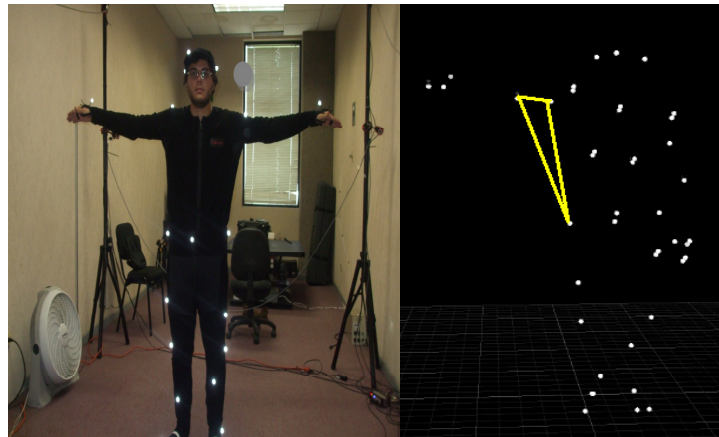


Figura 3. Captura de movimiento.

En la tabla 2, se muestran los ángulos obtenidos de una persona sana y un paciente, de acuerdo al segmento hombro, codo y muñeca.

Tabla 2. Resultados persona Sana y con Hemiplejía.

Paciente 1 Hemipléjico			ARTICULACIÓN	MOVIMIENTO	ÁNGULO
ARTICULACIÓN	MOVIMIENTO	ÁNGULO EN GRADOS	HOMBRO	FLEXIÓN	177.4337
HOMBRO	FLEXIÓN		HOMBRO	EXTENSIÓN	141.8856
HOMBRO	EXTENSIÓN		HOMBRO	ABDUCCIÓN	86.21
HOMBRO	ABDUCCIÓN		HOMBRO	ADUCCIÓN	11.84
HOMBRO	ROTACION MEDIA		HOMBRO	ROTACIÓN MEDIA	45.0254
HOMBRO	ROTACION LATERAL		HOMBRO	ROTACIÓN LATERAL	66.1072
HOMBRO	ADUCCIÓN	86.21	CODO	FLEXIÓN	45.1602
CODO	FLEXIÓN	70.3354	CODO	EXTENSIÓN	152.3405
CODO	EXTENSIÓN	136.6067	CODO	PRONACIÓN	62.4131
CODO	PRONACIÓN	20.237	CODO	SUPINACIÓN	0
CODO	SUPINACIÓN	62.4131	MUÑECA	FLEXIÓN	145.55
MUÑECA	FLEXIÓN	22.8999	MUÑECA	EXTENSIÓN	119.6595
MUÑECA	EXTENSIÓN	25.7955	MUÑECA	ABDUCCIÓN	167.143
MUÑECA	ABDUCCIÓN		MUÑECA	ADUCCIÓN	49.4704
MUÑECA	ADUCCIÓN				

Es necesario señalar que para lograr estos movimientos se realizaron tres ensayos indicándole los movimientos.

3.2. Segundo experimento

Análisis de movimiento de miembro superior derecho mediante guante de datos y phantom omni. En este estudio se propuso utilizar el protocolo PeaBody, que es una evaluación que se compone de seis subpruebas que miden las habilidades motoras interrelacionadas que se desarrollan a temprana edad. Está diseñado para evaluar las habilidades motoras de los niños desde el nacimiento hasta los 5 años de edad. Este protocolo es utilizado por los terapeutas ocupacionales, fisioterapeutas, especialistas en diagnóstico, intervención temprana, adaptados profesores de educación física, psicólogos, y otros que están interesados en examinar las habilidades motoras y que para este experimento fue utilizado en personas mayores [7].

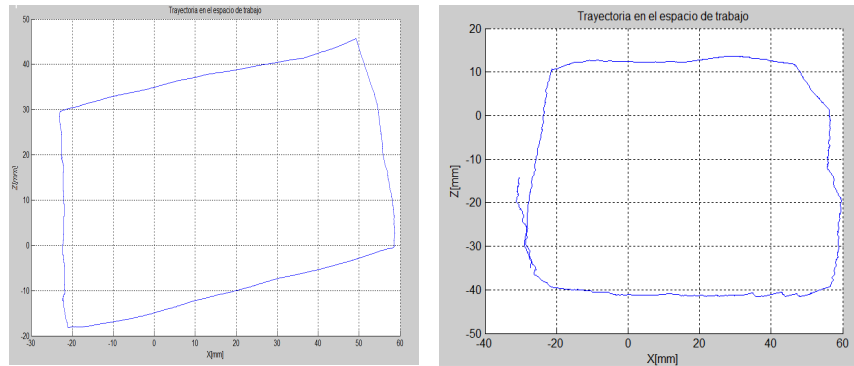
Dentro del grupo de subpruebas está la de tipo *Visual y Motor*, esta mide la capacidad de utilizar habilidades de percepción visual para realizar complejas tareas ojo-mano de coordinación, tales como alcanzar y agarrar un objeto, construir con bloques, y la copia de diseños[7]. Por lo que se diseñaron figuras como rectas paralelas horizontales y verticales, círculo, rectángulo y una cruz, para que el usuario realizara sobre la plantilla la copia del diseño y mediante el dispositivo háptico se capture como fue realizado el trazo, así mismo con el guante se mide el agarre en el efector final del dispositivo, para evaluar comportamiento de dedos, como se muestra en la figura 4, estas acciones son capturadas mediante las cámaras, lo que permite analizar el movimiento durante el proceso del trazo para evaluar el efecto que tendrá al usar dispositivos de guiado o control sobre los movimientos de personas hemipléjicas para el diagnóstico o rehabilitación.



Figura 4. Captura de datos dispositivos.

En esta fase los pacientes realizan la actividad estando sentados y mediante el phantom omni realizan el trazo sobre la plantilla, además se coloca la parte superior

del traje de captura de movimiento para un muestreo de los movimientos hechos en el miembro superior, como se observa en la figura 5.



a) Trazo persona sana b) Trazo persona hemipléjica
Figura 5. Trazo de rectángulo; a) persona sana b) persona hemipléjica.

Se analizó la diferencia entre la posición real y la posición deseada tomando como base el trazo hecho por una persona sana, como se muestra en la figura 6.

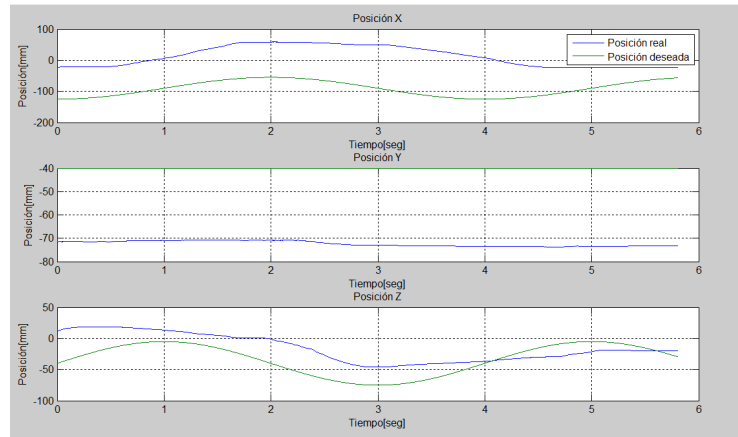


Figura 6. Análisis de posición al dibujar el rectángulo por la persona sana.

El movimiento en los pacientes sanos es constante y dibujan de izquierda a derecha desde la parte inferior en el caso de dibujar un rectángulo, sin embargo es problemático para una persona hemipléjica realizar este movimiento de abducción o aducción, sin embargo después intentos solo dos de las tres personas lograron realizar un movimiento dibujar el rectángulo.

Con el guante de datos de cinco sensores, se mide la flexoextensión del agarre del gimbal al realizar el trazo, permitiendo analizar el comportamiento por dedo, como se observa en la figura 7.a para personas sanas y figura 7.b en personas hemipléjicas.

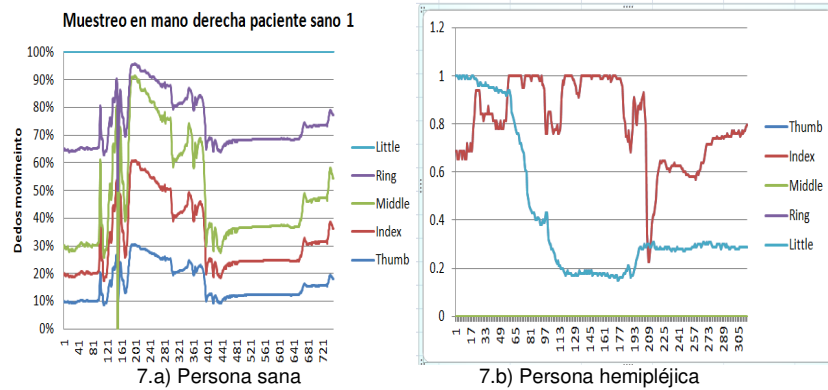


Figura 7. Análisis de agarre.

El agarre de los objetos de una persona hemipléjica es constante a través del dedo pulgar y meñique, esto debido a que al analizar la información se observa que existe espasticidad desde el comienzo del padecimiento, lo cual medicamente es recomendable atender en rehabilitación durante los primeros seis meses.

4. Resultados

Como parte de esta experimentación se comprobó con el volumen de datos almacenados, el movimiento angular del modelo D-H en pacientes sanos, permitiendo contar con la información mínima necesaria para establecer parámetros para posteriormente definir un umbral de movimiento que permita considerar al paciente hemipléjico como rehabilitado. En la figura 8, se muestran una gráfica comparativa entre los pacientes sanos y hemipléjicos, y con el resto de la información se tienen las siguientes observaciones:

1. Se verificó que el paciente hemipléjico presenta poco movimiento en hombro y dos de los tres casos presentaron dolor por hombro dislocado.
2. Existe una mayor amplitud angular en la circunducción en rotación lateral debido al desplazamiento de todo el miembro para tener agarre o movimiento de supinación; el origen del movimiento es la parte central media entre los miembros superiores, por lo que no es movimiento de hombro.
3. Los pacientes hemipléjicos tiene poca flexión en muñeca debido a que adquiere la mano la postura de extensión a causa de la espasticidad muscular.
4. Se obtienen rangos preliminares que sirven como referencia para el desarrollo algoritmos de rehabilitación con asistencia robótica.
 - a) En hombro se obtuvo un rango de 60° a 90° de movimiento de rotación.
 - b) El rango en el codo para supinación y pronación es de 48° a 63° , mientras que para la flexo extensión es de 49° a 145° .

- c) En la muñeca se tiene el rango de flexo extensión es de 25°-39°.
5. El uso de dispositivos robóticos sin algoritmos de control e implementando un protocolo, proporciona estabilidad física y mental, esto por comentarios de los pacientes siendo posible realizar una asistencia robótica para el diagnóstico y rehabilitación que tenga resultados óptimos.
 6. En los dedos se tiene un agarre en forma de pinza con dedo meñique y pulgar, con poco movimiento en dedo índice y anular. Es necesario implementar algún otro dispositivo o técnica que permita evaluar la abducción aducción.
 7. El diseño e implementación de algoritmos para guiado, fuerza y posición, requieren de una plataforma robótica robusta y con retorno de datos, que permita el desarrollo de la plataforma de diagnóstico y rehabilitación en miembro superior derecho y se estudia el uso del sistema Cyberforce, Cybergrasp y Dataglove 22 sensores.

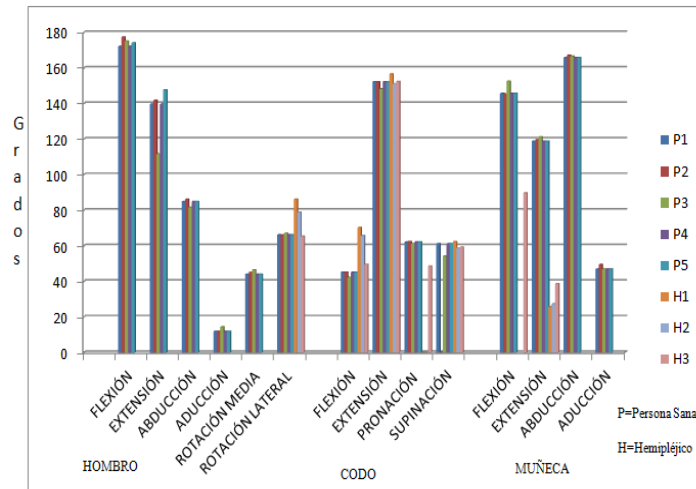


Figura 8. Resultado comparativo personas sanas y pacientes.

Como se puede observar en la gráfica se tienen movimientos comparativos entre personas sanas y pacientes, que permitirán tener rangos y umbrales para desarrollar sistemas asistidos con dispositivos robóticos.

5. Conclusiones

El uso de sistemas de captura de movimiento o sistemas con giroscopios, acelerómetros o con magnetómetro permiten una evaluación biomecánica sobre las actividades cotidianas, deportivas o de salud en los humanos. Sin embargo el campo de estudio requiere de un trabajo multidisciplinario para que exista una valoración y contribución en cualquier área. Para este estudio se contó con el apoyo del Centro de

Rehabilitación Integral zona Sur de Sinaloa, lo que permitió contar con el análisis del especialista en el área y contar con sugerencias dada la experiencia. A partir de estos datos se tienen rangos iniciales para el desarrollo de plataformas experimentales para asistencia robótica en la fisioterapia en miembro superior derecho.

Referencias

1. Rau, G., Disselhorst-Klug, C.: Movement Biomechanics Goes Upwards: From the Leg to the Arm. *Journal of Biomechanics*, 33, pp. 1207–1216 (2000)
2. Adamovich, S.V., Archambault, P.S.: Hand trajectory invariance in reaching movements involving the trunk. *Experimental Brain Research*, 138, pp. 288–303 (2001)
3. Optitrack, <http://www.naturalpoint.com>
4. Sensable, <http://www.sensable.com>
5. 5DT, <http://www.5dt.com/>
6. Pons, José: Wearable Robots, Biomechatronics Exoskeletons, Madrid, España, John Wiley. Robot and Human Interactive Communication, Toyama, Japón, pp. 57-67 (2009)
7. Provost, Beth: Concurrent Validity of the Bayley Scales of Infant Development II, Motor Scale and the Peabody Developmental Motor Scales-2 in Children with Developmental Delays. Research report Pediatric Physical Therapy, pp. 149-56 (2005)
8. Rymer: Rehabilitation Robotics. What Lies Ahead? In: Proc. of the International Neuro-rehabilitation Symposium 2009, pp. 1-20 (2009)
9. Seo: Design for presence: A structured Approach to Virtual Reality System Design (Teleoperators and Virtual Environments). Vol. 11, No. 4 pp. 120-126 (2002)
10. Spang: Robot Modeling and Control. First Edition, John Wiley & Sons, Inc.v(2004)
11. Popescu, V.G.: A Virtual Reality-based Tele-rehabilitation System with Force Feedback. IEEE Transactions on Information Technology in Biomedicine, vol. 4, issue, pp. 45 - 51, March (2000)
12. Boian, R.F.: Haptic Effects for Virtual Reality-Based Post-Stroke Rehabilitation. In: Proc. of 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 247-253 (2003)
13. Beth Provost: Concurrent Validity of the Bayley Scales of Infant Development II Motor Scale and the Peabody Developmental Motor Scales-2 in Children with Developmental Delays. Research Report, 0898-5669/04/1603-0149, Pediatric, Physical Therapy (2004)
14. Alamri, A.: Haptic Exercises for Measuring Improvement of Post-Stroke Rehabilitation Patients. International Workshop on Medical Measurement and Applications, Warsaw, Poland, May 2007, pp 1-10 (2007)
15. Chu, C.: Markerless Kinematic Model and Motion Capture from Sequences. In: Proceedings of IEEE Computer Vision and Pattern Recognition, pp. 475 (2003)
16. Mündermann: The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications. *Journal of Neuro-Engineering and Rehabilitation*, 3: 6-6 (2006)
17. Cyberglove systems, <http://www.cyberglovesystems.com/>
18. Beth: Assessment of motor development and function in preschool children. *Mental Retardation and Developmental Disabilities Research Reviews*, 11: 189–196 (2005)
19. Young, G: Assessment of Workload Using NASA Task Load Index in Perianesthesia Nursing. *Journal of PeriAnesthesia Nursing*, Vol 23, No 2 (April); pp. 102-110 (2008)
20. Sandra: Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research, pp. 1-46. NASA TLX Chapter1 (2003)

Pruebas de laberintos 2D y 3D con propósitos de uso en rehabilitación neuropsicológica

Víctor Tomás Tomás Mariano¹, Felipe de Jesús Núñez Cárdenas¹,
Raúl Hernández Palacios¹ y Mariano Pozas Cárdenas²

¹Universidad Autónoma del Estado de Hidalgo, Escuela Superior de Huejutla,
Huejutla de Reyes, Hidalgo, México
{victor_tomasm, felipe.huejutla, rpalacios}@hotmail.com

²Universidad Autónoma del Estado de Hidalgo, Centro de Investigación en Tecnologías de la
Información y Sistemas (CITIS),
Pachuca de Soto, Hidalgo, México
mpozas@hotmail.com

Resumen En el presente trabajo se muestran los resultados obtenidos del test de laberinto, al permitir recorrer un entorno virtual, los laberintos son ampliados según la configuración del especialista, estos laberintos permiten tener una mayor movilidad y direccionalidad al recorrerlos. Los laberintos generados son graficados en 2D y 3D; que satisfacen los indicadores valorados por especialistas médicos psicólogos para pruebas psicométricas. Una de las ventajas del sistema es que permite visualizar la solución del laberinto en 2D, se utilizan técnicas de búsqueda para encontrar la ruta-conexión entre dos celdas.

Palabras clave: laberintos ampliados, rehabilitación, algoritmos de búsqueda, graficación 2D y 3D.

1. Introducción

Los laberintos rectangulares son de los más utilizados en diferentes campos como lo es la robótica, educación, medicina y entretenimiento [1, 3, 17]. Existen varios algoritmos para generarlos [2], en este documento se realiza una descripción de los algoritmos más utilizados para la construcción de laberintos de conexión simple (LCS). También se generan LCS con pasillos ampliados en 2D y 3D, estos laberintos brindan una mayor movilidad y direccionalidad al recorrerlos, permitiendo tener laberintos de diferente complejidad, además de ser los que se utilizan en pruebas de rehabilitación. Se muestran los resultados obtenidos en la prueba de test del laberinto. El presente artículo está organizado de la siguiente manera: en el apartado 2, se resalta el uso de la realidad virtual en el proceso de rehabilitación y tipos de pruebas de laberintos; en el apartado 3 y 4 se habla de los tipos de laberintos y de los algoritmos para construirlos; en el apartado 5 se introduce a los laberintos ampliados y el problema que implica resolverlos mediante el algoritmo A*; en las secciones 6 y 7, se muestran los resultados obtenidos en laberintos 2D y 3D respectivamente; en el

apartado 8 y 9 se habla de las aplicaciones posibles y conclusiones generales del proyecto.

2. Laberintos y rehabilitación médica

La manera más común de construir laberintos es mediante un programa de computadora, y se ha puesto mucho énfasis sobre el potencial para el uso de ambientes virtuales en contextos médicos. Esta área se ha enfocado principalmente, en el tratamiento y rehabilitación de pacientes con alguna discapacidad psicomotriz [15, 16, 17].

El desarrollo de aplicación de los ambientes virtuales ha sido usado en la evaluación neuropsicológica (*Neural Assessment*) y rehabilitación cognitiva (*Cognitive Rehabilitation*) de componentes en procesos cognitivos incluyendo: *atención, memoria, habilidades espaciales y disfunciones del sistema nervioso central (Central Nervous System)* [9, 19, 24].

La evaluación neuropsicológica es una ciencia aplicada que evalúa cómo actividades específicas en el cerebro, son expresados en comportamientos observables, usa herramientas de evaluación psicométricas para el diagnóstico de disfunciones [15, 19]. La tecnología virtual brinda el potencial para desarrollar ambientes de prueba que pueden complementar los procedimientos de evaluación neuropsicológica existentes que dependen *principalmente de lápiz, papel y observación conductual*. Usado de esta manera, los ambientes virtuales para rehabilitación pueden mejorar la confiabilidad psicométrica y la validez puede causar una mejor detección y diagnóstico de las diferentes formas de disfunción del Sistema Nervioso Central [9]. Los ambientes de realidad virtual, constituyen una oportunidad interesante para la evaluación de la desorientación topográfica, proporcionando una representación de naturaleza dinámica e interactiva [18]. Entre los trabajos que se pueden mencionar está el de Nair et al, en donde la prueba del laberinto virtual, permite visualizar el recorrido realizado por un sujeto con discapacidad [22]. El procedimiento es comparado con un sujeto normal. Se puede observar que el medio por donde se realiza el recorrido, no necesariamente tiene la forma de un laberinto con paredes rectangulares, en vez de ello, se colocan obstáculos de diferentes formas dentro del ambiente [16].

Otro trabajo es el que presenta Bardorfer A. en donde el principal objetivo es hacer estudios funcionales de los miembros superiores (*Upper Limbs*) en pacientes con enfermedades neurológicas (Neurological Diseases). Los laberintos de prueba que se utilizan son 3D, de tamaño y ampliación de pasillo fijo. Entre los pacientes que desarrollaron la prueba se encuentran aquellos que tiene alguna disfunción psicomotriz, como lo es la enfermedad de Parkinson, Ataxia y Esclerosis Múltiple, para recorrer el laberinto, se utiliza una “palanca” que controla el movimiento de un “apuntador” (ball) usando una interfaz háptica PHANTOM [1, 16]. Un trabajo similar es el realizado por el Domínguez et al. [5] en lo referente a interfaces y sistemas de control aplicados en laberintos, utiliza un volante como interfaz háptica

para controlar el movimiento realizado por un operador humano con discapacidad motriz en un laberinto virtual 3D.

Los trabajos antes mencionados consideran el uso de laberintos para rehabilitación en ambientes virtuales, estos utilizan laberintos prefijados con distintos niveles de complejidad en su estructura, pero no generan una solución automática del laberinto, en vez de ello, la solución se realiza por una persona sana o por el especialista de manera manual.

2.1. Pruebas psicométricas

Test Laberintos de Porteus (TLP): Fue diseñado en 1914 por S.D.Porteus, siendo propuesto originalmente como una prueba capaz de medir la inteligencia general de una persona en términos de edades mentales. La prueba consiste en la resolución de laberintos ordenados en un modelo de dificultad creciente, debiendo la persona trazar con un lápiz el camino desde la entrada hasta la salida cumpliendo consignas que permiten ubicarlo entre las pruebas de funciones de planificación. El TLP es empleado actualmente como un test asociado a la activación de la región frontal del cerebro, implicado en el factor de planeamiento y capaz de detectar errores de tipo perseverantico [12].

Laberintos de Foster. El laberinto de Foster se utiliza para evaluar destrezas manuales, memoria y aprendizaje. La tarea consiste en terminar el recorrido del laberinto según los criterios de la investigación: utilizando el menor tiempo posible, en transferencia de aprendizaje entre las manos, cometiendo el menor número de errores por ensayo, con o sin retroalimentación. Este ejercicio puede evaluar la memoria espacial y la solución de problemas. Las áreas en que se emplea son: aprendizaje, procesos cognoscitivos y senso-percepción. Generalmente, tanto la velocidad del trazado como los callejones de recorrido, son registrados. Típicamente se obtiene una curva de aprendizaje negativa, y la tasa de aprendizaje a través de varios ensayos, puede ser analizada. El laberinto ha sido diseñado para que pueda contabilizar automáticamente el número de errores y el tiempo total para completar el recorrido [12].

3. Tipos de laberintos

Laberinto de Conexión Simple (LCS). Son aquellos que tienen puntos muertos o callejones sin salida, por lo tanto, solo tienen una solución entre la entrada y la salida, en el proceso de buscar la solución, en ocasiones, se recorre gran parte del laberinto, causando frustración en el espectador [2].

Laberinto de Conexión Múltiple (LCM). Estos laberintos tienen más de una solución desde la entrada a la salida. En este tipo de laberintos, además de encontrar una solución, se requiere que sea la más corta. Tienen ciclos internos, lo que dificulta aún más el proceso de buscar la solución, ya que se debe tomar en cuenta en que secciones del laberinto ya se ha transitado, evitando dar giros en un mismo segmento del laberinto [2].

4. Construcción de laberintos

La construcción al azar permite generar gran cantidad de laberintos conteniendo corredores como las ramas de un árbol, persiguiendo la desorientación del explorador, siendo a su vez ésta la forma más fácil de implementarse en un programa de computadora. Los algoritmos explicados a continuación, generan laberintos LCS completamente conectados [2].

La construcción al azar se basa en dos enfoques, los cuales se enuncian a continuación:

- **Cavar túneles.** Como su nombre lo indica, se refiere a cavar túneles a lo largo y ancho del laberinto hasta cubrirlo en su totalidad, como la exploración de una mina. A este tipo de construcción, pertenecen:
 - Algoritmo de construcción Kruscal.
 - Algoritmo de construcción Aldous-Broder.
 - Algoritmo de construcción Prim's.
 - Algoritmo de construcción Recursivo Bactracker.
- **Levantar muros.** Consiste en elevar los muros de los corredores por todo el cuerpo del laberinto, como en la construcción de un edificio.

4.1. Algoritmo de Kruscal

Este algoritmo construye laberintos de conexión simple completamente conectados, levanta muros.

1. Especificar el tamaño o las dimensiones del laberinto de acuerdo con las habitaciones que se desea que contenga.
2. Dividir el laberinto en habitaciones en base a sus dimensiones y etiquetarlas con una identificación única.
3. Seleccionar una habitación al azar y cavar un túnel hacia una habitación adyacente (en caso de existir más de una, elegir al azar) sólo si tiene diferente etiqueta, y etiquetar la habitación o habitaciones que se unieron con la identificación de la habitación inicial; repitiendo este proceso hasta que todas las habitaciones tengan la misma etiqueta.

Marcar la ubicación de la entrada o del punto inicial de la exploración y la ubicación de la salida.

4.2. Algoritmo Aldous-Brother

Es una técnica simple, pero puede llevar gran tiempo la construcción de un laberinto, esto es porque se va moviendo al azar dentro del laberinto sin discriminar las habitaciones con las que ya se cavaron túneles. Los pasos se enuncian a continuación:

1. Especificar el tamaño o las dimensiones del laberinto de acuerdo con las habitaciones que se desea que contenga.
2. Dividir el laberinto en habitaciones en base a sus dimensiones.
3. Elegir al azar una habitación dentro del laberinto.
4. Moverse a una habitación vecina adyacente al azar y cavar un túnel hacia la habitación anterior en el caso de que la nueva habitación no sea parte de otro túnel, continuando con este paso hasta que todas las habitaciones sean parte del laberinto.
5. Marcar la ubicación de la entrada o del punto inicial de la exploración y la ubicación de la salida.

Por motivos de espacio no se describen los demás algoritmos, pero el resultado de estos siempre es un laberinto de conexión simple con una sola pista

5. Desarrollo

5.1. Introducción a laberintos ampliados

La representación matricial en “0” y “1”, del laberinto final de cada algoritmo de construcción; permite representar las celdas libres en blanco, celdas ocupadas de color negro, para indicar la posición (i, j) dentro del laberinto se utiliza en su totalidad la celda respectiva de la matriz, en la fig. 1 se indica con un color distinto la entrada E y salida S del laberinto.

Se puede aumentar la ampliación, para obtener nuevos laberintos ampliados, independientemente del algoritmo de construcción aplicado, lo que es una ventaja, ya que permite conocer la estructura del laberinto y por lo tanto la orientación en la ubicación de la salida, además, los laberintos ampliados tienen mayor movilidad permitiendo tener más direccionalidad respecto a los laberintos de una sola vía, ver fig. 1.

5.2. Búsqueda de ruta solución en laberintos ampliados

El problema de búsqueda de rutas, también conocido como *maze problem*, se ha discutido en varios trabajos de investigación [6, 7, 10, 23] para el caso de estudio en particular, se tiene un laberinto con múltiples vías o laberinto ampliado, en el que, dependiendo de las dimensiones y complejidad, el número de rutas posibles a encontrar dentro del laberinto se incrementa, ver fig. 1.

Un factor importante de las rutas posibles, depende de la ampliación k que tiene el laberinto, además, una vez abandonado el punto de entrada, el número de combinaciones posibles en los movimientos puede llegar a ser muy grande, lo que provoca que el explorador visite una mayor cantidad de posiciones en el laberinto, ya que por cada celda expandida pueden generarse otras más según la cantidad de vecinos transitables que ésta tenga, tal como se muestra en fig. 1, por lo que es necesario de algún modo “recordar” que secciones del laberinto ya han sido exploradas, y así, tomar la mejor decisión de movimiento en el espacio de búsqueda.

Visualizar un laberinto como un grafo, permite retomar el “comportamiento” de los algoritmos principales: Depth First Search (DFS) [4], Breadth First Search (BFS) [4], Dijkstra [11] y su posible implementación para encontrar una solución, de los que, el segundo y el último son óptimos pero el espacio en memoria necesario puede volverse crítico, mientras que el primero de ellos no promete una trayectoria con costo mínimo.

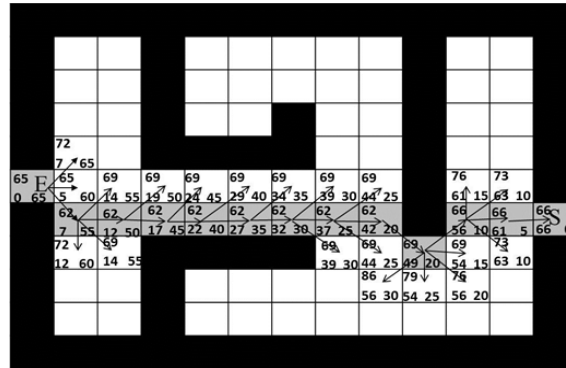


Fig. 1. Trayectoria óptima que enlaza la entrada y la salida del laberinto utilizando A*.

Otros algoritmos permiten modelar el problema en entornos tipo rejilla, y utilizar una matriz A de $m \times n$, en los que se requiere procesar datos dentro de una matriz binaria, ceros y unos, con la finalidad de identificar áreas libres y obstáculos en el laberinto. Entre los algoritmos analizados están principalmente: Algoritmo de Lee [13] [21] y Moore [14], Algoritmo de Hadlock [20], Algoritmo de Soukup [20], Algoritmo A* [8], siendo este último el elegido para ser implementado en un laberinto ampliado ya que la cantidad de nodos que explora es menor que cualquiera de los anteriores, y por garantizar la ruta más corta.

A modo de ejemplo, ver fig. 1, se utiliza un laberinto de ampliación $k=2$, con movimientos en horizontal, vertical y diagonal, donde éste último genera un costo más alto que los dos primeros, con fundamento en el teorema de Pitágoras. En el que para un movimiento en horizontal o vertical se asigna una magnitud de 5 unidades, generando entonces que el costo en diagonal se aproxime a 7 unidades, los cuales son tomados para trazar la ruta más corta en el laberinto, donde la celda E representa el punto de partida, la celda S el punto objetivo o meta, las celdas libres representan caminos posibles y las celdas de color negro representan los obstáculos.

6. Resultados en 2D

La representación de cada celda al graficar en pantalla, varía desde el valor mínimo de 1 pixel hasta m pixeles. Una de las características del sistema desarrollado es que permite almacenar los laberintos, dando la posibilidad de volver a “reproducir” una prueba de interés y ver el comportamiento del usuario durante el recorrido, ver fig. 2.

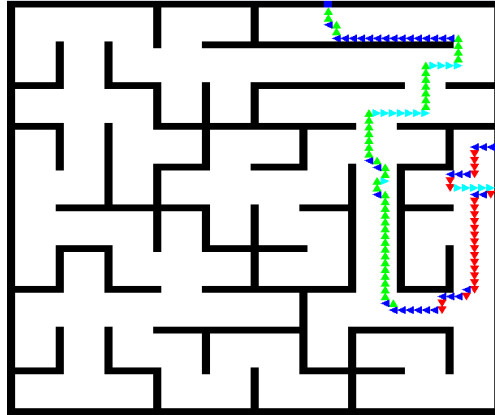


Fig. 2. Laberinto con $k=5$, celda = 10 píxeles.

En la fig. 2, se visualiza el recorrido realizado por el usuario desde el punto inicial con un cuadro de color rojo, y con un cuadro de color azul el punto final, los movimientos se visualizan de la siguiente manera: derecha de color azul, abajo de color rojo, a la izquierda de color cian, y arriba de color verde. Se muestra la posición actual del usuario, el movimiento realizado se resalta al cambiar el color del indicador, observe que se puede girar en una sección muy pequeña del laberinto ampliado.

Al finalizar el recorrido del laberinto, se muestra un mensaje informativo del tiempo de duración en llegar al final, una gráfica de latencias, que mide el tiempo que tarda el explorador en elegir el movimiento a realizar, total de movimientos efectuados y los choques realizados por el explorador en el recorrido del laberinto.

7. Resultados 3D

Una de las ventajas del recorrido de un laberinto en 2D, es que permite visualizar la salida del laberinto, que otorga al explorador la posibilidad de orientar sus movimientos de búsqueda hacia la salida. Sin embargo, para el recorrido en 3D, el proceso es más complicado, debido a que la vista del laberinto, se hace en una sección muy pequeña del mismo, esto desorienta al usuario.

En la interfaz 3D se tiene una pantalla principal con 2 cámaras: una delantera, parte izquierda y una trasera, parte superior derecha, además para ayudar al usuario, se muestra el laberinto en 2D, parte inferior derecha a fin de posibilitar su ubicación en el laberinto, ver fig. 3. Una vez abandonado el punto inicial, el explorador se interna en el laberinto teniendo una vista pequeña del mismo, ver fig. 3, esto hace que se pueda perder en una sección limitada del mismo.

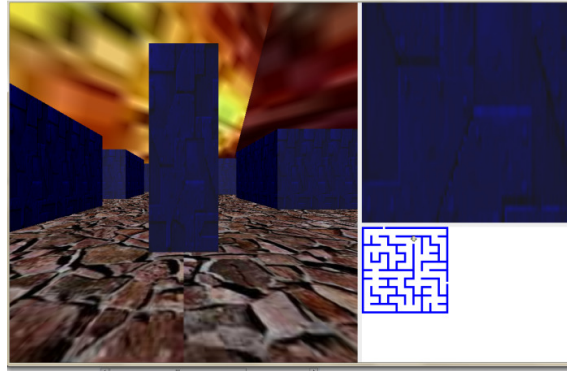


Fig. 3. Vista Interna al Recorrer el Laberinto en 3D.

8. Aplicaciones

Una de las aplicaciones de los laberintos es en la rehabilitación neuropsicológica, los laberintos son pruebas de gran utilidad para la detección de déficit de atención y síndrome frontal. Permiten evaluar diferentes niveles de integración nerviosa y trastornos específicos como secuela de una lesión cerebral [5], [9], [15].

La robótica es una de las áreas en donde el problema de planeación de trayectorias en ambientes o espacios abiertos dinámicos tiene gran aplicación, en ocasiones se cuenta con obstáculos extras en el camino, en estos casos, el robot se auxilia de sensores para su detección y con la implementación de algoritmos que brindan información local, se puede encontrar una solución [3], [6], [7], [10].

El conocimiento de la estructura del laberinto por donde se mueve el robot es de gran importancia ya que se debe tomar en cuenta la forma del propio robot para la planeación de trayectorias. En este sentido, se profundizó en los algoritmos que se aplican a laberintos de estructura conocida [2], [20]. Una posible mejora es abarcar situaciones que permitan modelar el problema cuando la salida y estructura del laberinto es desconocido, así mismo, se pretende llevar un “aprendizaje” del recorrido realizado.

Para la Graficación en 3D se hace uso del código implementado en el libro *Killer Game Programming in Java* By Andrew Davison, capítulo 25.

9. Conclusiones

Los resultados arrojados por el sistema permiten generar laberintos ampliados de diferentes tamaños y complejidades, totalmente conectados y perfectos. Las pruebas se guardan en un sistema de archivos para recuperar información.

El proceso de buscar una solución en un laberinto ampliado es más complejo, se tienen diferentes rutas que conectan la entrada y salida; de los algoritmos analizados

se optó por implementar el algoritmo A* para encontrar la ruta óptima en el laberinto ampliado. Garantizando que la solución dada sea segura, es decir, libre de colisiones.

Como trabajo futuro, se pretende medir la diferencia entre el recorrido realizado por el usuario y la solución generada por el sistema con el objetivo de medir el grado de error presentado en secciones del laberinto.

Referencias

1. Bardorfer A., Munih M. Zupan A. and Primožic A., "Upper Limb Motion Analysis Using Haptic Interface", IEEE/ASME Transactions on Mechatronics, Vol. 6, No. 3, September 2001.
2. Bernabe S. E. "Construcción y Recorrido de Laberintos", Pachuca de Soto, Hidalgo, México, Julio 2004.
3. Buckley S. J., "Fast Motion Planning for Multiple Moving Robots", IBM T. J. Watson Research Center, 1989.
4. Cormen T. H., Charles E. L., Rivest R. L., and Clifford S. "Introduction to Algorithms", Second Edition. MIT Press and McGraw-Hill, Section 23.2: The Algorithms of Kruskal and Prim, pp.567-574, 2001.
5. Domínguez R. O. A., López M. V., Samperio L. R., "Resultados Preliminares sobre Interacción Háptica en Laberintos Virtuales, con Propósitos de Diagnóstico en Pacientes con Discapacidades Neuropsicológicas", Tesis, UAEH, CITIS, Hidalgo México, 2005.
6. Dracopoulos. D. C., "Robot Path Planning for Maze Navigation", Brunel University, Department of Computer Science, IEEE, 1998.
7. Gordon V. S., Matley Z., "Envolving Sparse Direction Maps for Maze Pathfinding", California State University, Sacramento and Digital Eclipse Software, Inc. Vancouver, 1992.
8. Hart, P. E., Nilsson, N. J., & Raphael, B. (1964). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, SSC-4 (2).
9. Huiyu Z. & Huosheng H., "Inertial Motion Tracking of Human Arm Movements in Stroke Rehabilitation", Proceedings of the IEEE International Conference on Mechatronics & Automation, Niagara Falls, Canada, July 2005.
10. Jan, G. E., & Chang K-Y, P. I. (2003). A New Maze Routing Approach for Path Planning of Mobile Robot. IEEE International Conference on Advanced Intelligent Mechatronics.
11. Johnsonbaugh, R. (1999). Teoría de Gráficas: Un algoritmo para la ruta más corta. En Matemáticas Discretas. México: Prentice Hall.
12. Kaplan, R. M., & Saccuzzo, D. P. "Psychological Testing: Principles, Applications, and Issues". Thomson Wadsworth. 2005.
13. Lee C. Y., "An Algorithm for Path Connections and its Applications", IRE Transactions on Electronic Computers, vol. EC-10, pp. 346-365, September 1961.
14. Moore E. F., "The Shortest Path Through a Maze", Annals of Computation Laboratory of Harvard University, Harvard University Press, vol. 30, pp. 285-292, 1959.
15. Morgante F., Gaggioli A. Strambi L., Rusconi M.L. and Riva G., "Computer-enhanced Route and Survey Spatial Knowledge Assessment in Clinical Neuropsychology", IEEE Technology for Neuro-Psychology Lab, Istituto Auxologico Italiano, 2006.
16. Nair P., Jadhav C., Krivi V., "Development and Testing of a Low-Cost Diagnostic Tool for Upper Limb Dysfunction", Department of Mechanical and Aerospace Engineering, State University of New York at Buffalo, Buffalo, NY.

17. Reinkensmeyer D. J. Pang C. T., Painter C. C., "Web-Based Telerehabilitation for the Upper Extremity After Stroke", IEEE Transactions of Neural Systems and Rehabilitation Engineering, Vol. 10, No. 2, June 2002.
18. Rizzo A., Buckwalter J.G., van des Zaag C., Neumann U., "Virtual Environment Applications in Clinical Neuropsychology", IEEE, University of Southern California, Fuller Graduate School of Psychology, Technology for Neuro-Psychology Lab, 2005.
19. Rose F.D., Attree E.A., Brooks B.M., Johnson D.A., "Virtual Environments in Brain Damage Rehabilitation: Rationale from Basic Neuroscience", Department of Psychology, University of East London, London, UK., 1998.
20. Sait, S. M., & Youssef, H. (1999). En VLSI Physical Design Automation: Theory and Practice (págs. 233-276). Singapore: World Scientific Publishing Co.
21. Sheldon A. B., "A Modification of Lee's Algorithm", IEEE, Transactions on Electronic Computers, Febrero, 1967.
22. Stefani O. Mager R., "Cognitive Ergonomics in Virtual Environments: Development of an Intuitive and Appropriate Input Device for Navigating in a Virtual Maze", Applied Psychophysiology and Biofeedback, Vol. 30, No. 3, September 2005.
23. Sutherland E. I., "A Method for Solving Arbitrary-Wall Mazes by Computer", IEEE Transactions on Computers, vol. c-18, no. 12, December 1969.
24. Wann J. P., Rushton K. S., Smyth M. and Jones D., "Rehabilitative Environments for Attention and Movement Disorders", Communications of the ACM, vol. 40, no. 8, Agosto 1997.

Desarrollo de un lápiz óptico inalámbrico para aplicaciones de rehabilitación motriz fina

Roberto A. Montaña-Murillo, Rubén Posada-Gómez, Albino Martínez-Sibaja,
Patricia I. Cornelio-Martínez, Blanca E. González-Sánchez
y Alberto A. Aguilar-Lasserre

División de Estudios de Posgrado e Investigación,
Instituto Tecnológico de Orizaba,
Orizaba, Veracruz, México

```
{ingeniero_montano, paty_corneli, blancags67, albertoaal}  
@hotmail.com, rposada@itorizaba.edu.mx,  
albino3_mx@yahoo.com
```

Resumen La discapacidad motriz es una de las afecciones de movimiento más comunes, debido a que cualquier persona en cualquier etapa de su vida puede ser afectada de forma parcial o total, temporal o permanente, cambiando no sólo su movilidad o coordinación, sino su forma de vida. Por esto, es que se hacen necesarios equipos que ayuden a mejorar el índice de recuperación en este tipo de pacientes sometidos a rehabilitación motriz, principalmente niños. En este trabajo se propone un equipo auxiliar a las terapias de rehabilitación motriz con base en movimiento activos, mediante ejercicios en las extremidades superiores, frente a un sistema de captura de imágenes, realidad aumentada y retroalimentación audio-vibro táctil, mediante lápiz óptico inalámbrico con comunicación por radiofrecuencia.

Palabras clave: rehabilitación extremidades superiores, dibujo 2D, motricidad fina, retroalimentación vibro táctil, lápiz óptico, realidad aumentada.

1. Introducción

En México el 5.1% de la población sufre de algún tipo de discapacidad [1], siendo la discapacidad motriz, la más común, ya que el 58.3% del total de personas con discapacidad, padecen este tipo de problemas motrices. Las discapacidades de tipo motriz, son aquellas afecciones de movimiento y coordinación. Se dividen en dos grupos, la discapacidad motriz gruesa y la discapacidad motriz fina, al primer grupo pertenecen los problemas de movimiento relacionados con fuerza, velocidad y equilibrio, en actividades tales como: caminar, correr, saltar, gatear, etc. En el segundo grupo se encuentran los problemas en donde se necesita de una coordinación y precisión elevadas, dificultando la realización de acciones como la escritura, punzar, enhebrar y movimientos en los que se necesita una buena coordinación visomanual [2].

El proceso de rehabilitación motriz se basa en el concepto de neuroplasticidad, este concepto ofrece la posibilidad de mejorar el rendimiento y las capacidades neuronales del paciente, y también es considerado como la base biológica que apoya la rehabilitación de las funciones cognitivas que se pierden debido a un daño cerebral, se define como la ciencia que estudia el conjunto neuronas, la estructura de sus redes y su función mediante las nuevas experiencias. Ésta se obtiene a través de rutinas de rehabilitación que están fundamentados en la repetición de una actividad motora, con el objetivo de transmitir un estímulo a los pacientes y realizar una reasociación de neuronas a los movimientos de las extremidades afectadas [3].

La rehabilitación para la discapacidad motriz, conjunta ejercicios pasivos (ejercicios en los que el paciente no realiza movimientos voluntarios del musculo afectado) como activos (el paciente realiza movimientos para ejercitar el musculo o la extremidad afectada de forma voluntaria) [4], con la finalidad de mejorar la condición o el estado del paciente, así como tratar de integrarlo a las actividades sociales cotidianas.

Cabe señalar que tanto el procesamiento de imágenes como el uso de retroalimentación vibracional convencionalmente se emplean en sistemas de realidad virtual para mejorar la calidad de interacción del usuario con el sistema [5], tanto en ambientes bidimensionales, como la tele manipulación en 3D [6], sin embargo, el procesamiento de imágenes es una de las herramientas que recientemente ha sido utilizada con el fin de ayudar a la rehabilitación de los pacientes. Ya sea analizando las imágenes médicas de diferentes lesiones [7], o empleando marcadores pasivos y activos para supervisar la recuperación y mejorar la movilidad en el entorno familiar de pacientes con accidentes cerebrovasculares y sistemas de mayor complejidad que incluyen rehabilitación mediante sistemas robóticos [4].

Por otra parte, se han publicado diversos estudios que sostienen que los estímulos vibratorios durante el proceso de rehabilitación generan excelentes resultados, y no solo como terapia muscular u ósea mediante el principio de contracción muscular refleja [8], o para proporcionar un incremento sensorial [9], sino también como retroalimentación hacia el usuario, indicando acierto o error en los ejercicios realizados durante las terapias que conforman el proceso de rehabilitación [10], a ésta se le llama retroalimentación vibro táctil.

Hablando del procesamiento de imágenes enfocado a la rehabilitación, se pueden encontrar la implementación de mesas para la rehabilitación de las extremidades superiores, mediante el procesamiento de imágenes adquiridas desde una cámara, posicionada sobre un eje perpendicular la mesa, en la superficie de la mesa se desplegaban un conjunto de trayectorias, y utilizando un marcador pasivo se definen los movimientos del usuario [11].

Existen diversas publicaciones acerca de la retroalimentación vibracional, enfocada a los pacientes con problemas de vestibulopatía, mediante sensores de inclinación y retroalimentación para la corrección de la postura [12]. En un trabajo similar, se emplean acelerómetros triaxiales de un celular para evaluar la inclinación del paciente, accionando los impulsos vibro táctiles desde el conector de auriculares del teléfono celular [13], experimentos propuestos para este tipo de equipos se realizaban mediante una plataforma de movimientos aleatorios [14].

Los equipos de rehabilitación han cambiado mucho en los últimos años, cada vez son más numerosos los sistemas que integran impulsos vibratorios, luminosos o auditivos, procesamiento de imágenes, ambientes virtuales, realidad aumentada,

implementación de movimientos como medio de retroalimentación entre el sistema y el usuario. Las aplicaciones para este conjunto de técnicas (procesamiento de imágenes, realidad virtual, realidad aumentada y retroalimentación vibro táctil) se han hecho muy diversas y aplicables.

Existen diversos equipos que integran el procesamiento de imágenes mediante marcadores activos para evaluar el rango de movimiento en pacientes sometidos a rehabilitación de brazo, e interactuando con actividades enfocadas a mejorar el rango de movimiento de la parte afectada [15], pero los marcadores luminosos, no son la única manera de interactuar con ambientes virtuales de rehabilitación, también es posible encontrar sensores y actuadores magnéticos para corregir el factor de error mediante estímulos vibratorios [16].

Otra de las aplicaciones de los impulsos como medio de retroalimentación es desarrollo de un robot planar, este se enfoca en la rehabilitación de los miembros superiores del cuerpo, mediante el uso de brazos robóticos y en función de los estímulos vibro táctiles, corrige errores en las trayectorias seguidas por los usuarios del equipo [10]. La realidad virtual permite al usuario interactuar con elementos dentro de un escenario simulado. Recientemente, están realizando proyectos, equipos y dispositivos basados en realidad virtual aumentada, como una de las novedades más relevantes en la neurorrehabilitación [17].

En la actualidad esta técnica está causando un fuerte impacto en el área de rehabilitación con proyectos enfocados al tratamiento de problemas de discapacidad motriz de los miembros superiores [18] e inferiores del cuerpo.

Este sistema se basa en la rehabilitación por movimientos activos, y por esto, para este trabajo, se desarrolló un equipo que opera de manera remota, y mediante el uso de un lápiz óptico inalámbrico como marcador activo e instrumento de sensado de fuerza y en conjunto con un procesamiento de imágenes, se logra interactuar con el usuario mediante dos modalidades de uso, la primera contiene ejercicios de seguimiento de trayectorias o figuras patrón visualizadas en la interfaz, el segundo modo de uso, es la función de dibujo libre, en la cual el paciente tendrá un área efectiva de dibujo de un metro veinte centímetros para realizar trazos y dibujos que permitan al paciente un periodo extra en el uso del equipo.

La forma de interacción usuario-interfaz es como ya se mencionó, mediante la función de dibujo, realizando los trazos al aire frente al sistema de captura de imágenes y utilizando retroalimentación audio-vibro táctil, basado en comunicación por radiofrecuencia para reducir el error en los trazos del usuario, y utilizando el sensado de fuerza como medio para establecer un umbral de activación y desactivación de la función de dibujo así como seleccionar el color de los trazos a realizar.

El sistema evalúa no solo las incidencias del trazo realizado por el usuario con la imagen propuesta en el ejercicio, sino también la velocidad y precisión en que se realizan las trayectorias en la sesión de rehabilitación de los miembros superiores.

En este trabajo se muestra el diseño del lápiz óptico, sus características y las tareas que de las que se encarga, también se muestra el diseño de la interfaz gráfica y el procesamiento que realiza para la identificación del marcador activo que se encuentra en el lápiz óptico, así como también, se presentan los resultados de un periodo de pruebas con pacientes en proceso de rehabilitación motriz por lesión y el análisis de los datos obtenidos en las pruebas.

2. Lápiz óptico

La primera contiene el hardware de proyecto (lápiz óptico), en la cual, el elemento principal es un microcontrolador PIC16F628A encargado de recibir las lecturas de los sensores de fuerza, como de regular tanto la activación como la desactivación de las señales luminosas, auditivas y vibratorias, mediante las palabras de control recibidas por la interfaz del proyecto, la comunicación se establece por radio frecuencia, mediante módulos Xbee serie 2, estos módulos trabajan con una frecuencia de operación de 2,4 GHz, con un consumo de corriente de 40mA tanto en recepción como en transmisión y una distancia de 40-120m, además de su bajo consumo, estos módulos son de bajo costo, versátiles y sencillos de usar en sus diversas topologías. El protocolo utilizado para la comunicación de los módulos, es el protocolo “Zigbee”, en una red punto a punto, con topología estrella. En la figura 1 se muestra el diagrama electrónico del circuito de esta etapa así como el prototipo de lápiz óptico.

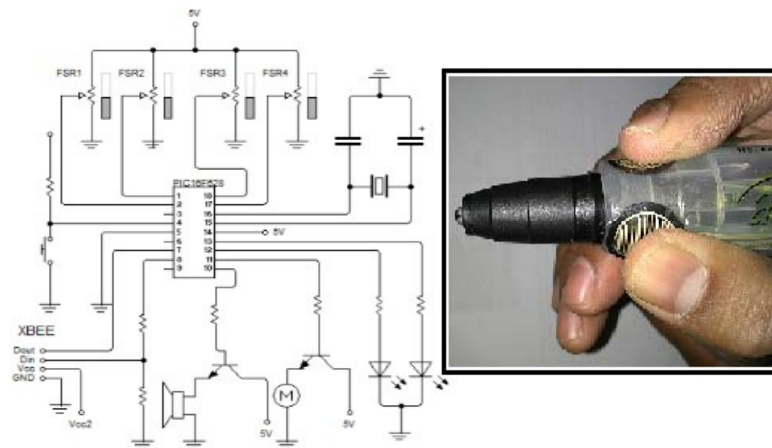


Fig. 1. Diagrama eléctrico del lápiz óptico e imagen del prototipo de lápiz óptico.

En la figura 2 se muestra el diagrama de flujo de la operación del microcontrolador que rige el hardware del proyecto, comenzando con la inicialización del sistema y comprobando si la comunicación con la interfaz está en funcionamiento, establecida la comunicación, se comienza con la lectura de los sensores FSR (Force Sensor Resistor), una vez realizadas estas acciones el sistema realiza la primera condición IF, si el dato recibido por la comunicación es “Encender” entonces el equipo encenderá el marcador activo posicionado en la punta del lápiz óptico, permitiendo con esto que la interfaz determine una posición para el marcador, el lápiz óptico, envía las lecturas de los sensores de fuerza posicionados en los puntos clave para la manipulación del lápiz, ahora el sistema queda en espera a una nueva palabra de control para manipular las señales sensoriales con las que cuenta el lápiz, para establecer error o acierto en el trazado de las trayectorias del usuario con respecto de patrón desplegado por la interfaz.

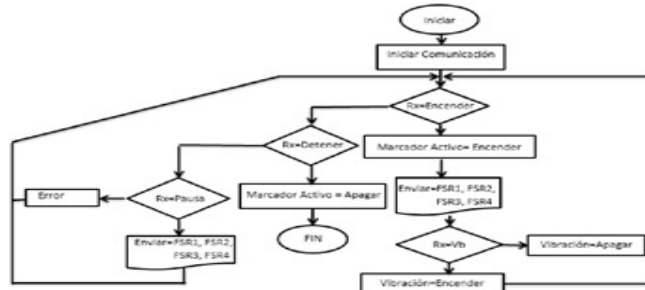


Fig. 2. Diagrama del prototipo.

3. Interfaz gráfica

La segunda etapa de proyecto es la interfaz gráfica (figura 5), ésta se encarga de la captura y procesamiento de las imágenes para la ubicación del marcador y la lectura de los valores de los sensores FSR recibidos por la etapa anterior, ya que es en función de estos, que se evaluará el momento en que inicia la función dibujo y los cambios de color en las líneas a trazar, el sistema cuenta con 4 umbrales, si el valor promedio de los sensores no supera el valor de ninguno de los umbrales, la interfaz permanecerá solo monitoreando los movimientos del lápiz óptico inalámbrico. El primero es el umbral mínimo, el cual cuando es superado, se activa el modo dibujo, y el sistema sale del reposo y plotea la trayectoria por la que se desplaza el marcador activo, el color de la línea ploteada es amarillo, el segundo es el umbral de color1, cuando este umbral es superado el color de la trayectoria a partir del punto en el que se encuentre cambia a rojo, el tercer umbral es de color2, y de la misma manera que el umbral anterior, cuando es superado, la línea ploteada cambia de color pero ahora el color asignado es azul, y finalmente el ultimo umbral torna a la línea ploteada a un color negro.

Esta interfaz está diseñada en la plataforma de Matlab, empleando el toolbox de adquisición de imágenes.

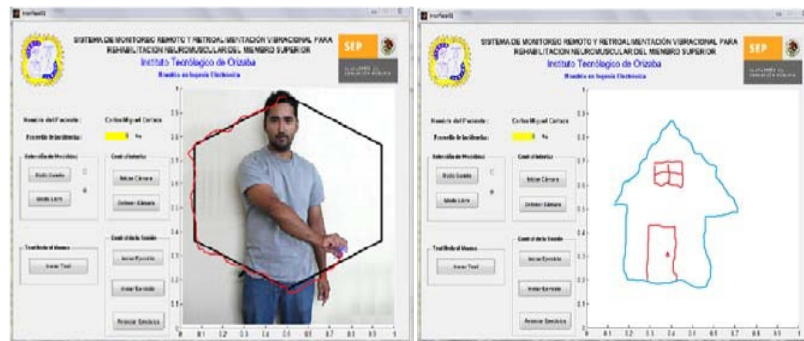


Fig. 3. Interfaz gráfica en ejercicio con realidad aumentada y modo libre.

Una vez inicializada la interfaz, se espera a que se introduzca el nombre de usuario o paciente para poder guardar los datos de la sesión en el registro del paciente, y poder consultarlos posteriormente en la base de datos, una vez introducido el nombre del paciente, se oprime el botón de inicio, para comenzar con la sesión, en este momento es cuando se manda la activación a la etapa del lápiz óptico, para el encendido del marcador activo, y es cuando se inicia la recepción de los datos de los sensores de fuerza, evaluando el promedio de estos con los valores de los umbrales ya mencionados, y graficando según se superen los umbrales, al mismo tiempo se inicia la captura de imágenes y el procesamiento para la ubicación del marcador activo (lápiz óptico inalámbrico).

El sistema cuenta con dos modos de operación, el primero es el modo de dibujo guiado, en el cual se muestran trayectorias o figuras predefinidas en la interfaz para que el usuario las siga y en función de la comparativa del trazo del usuario y la figura patrón, se activaran las señales vibratorias indicando error o no coincidencia en la comparación o un sonido intermitente indicando acierto, es en este modo en el que se utiliza realidad aumentada durante la sesión de rehabilitación. El segundo modo de operación es el dibujo libre, donde el usuario cuenta con un espacio de 1.20m x 1.20m para realizar dibujos de forma autónoma.

La adquisición de las imágenes se efectúa con una cámara web con una resolución máxima de 1.3 Megapíxeles y una velocidad de captura de 30 fps. Para aumentar la velocidad de captura y procesamiento de las imágenes y que el sistema pueda operar en tiempo real, la captura de imágenes se hace con una resolución 320 x 240.

Una vez que se ha adquirido la imagen actual, se efectúa el procesamiento de la imagen. Para ello, se aprovechan las características del marcador activo, es decir su geometría y color. De esta manera, puesto que se emplea un led azul ultra brillante, se hace una identificación del color en la matriz Azul de la imagen RGB adquirida originalmente. Adicionalmente se convierte con ayuda de la ecuación (1) la imagen RGB al formato HSV para identificar los niveles de brillo dados en la matriz V [19].

$$H = \begin{cases} \frac{g-b}{\max(r,g,b)-\min(r,g,b)}, & \text{si } r = \max(r, g, b) \\ \frac{b-r}{\max(r,g,b)-\min(r,g,b)} + 2, & \text{si } g = \max(r, g, b) \\ \frac{r-g}{\max(r,g,b)-\min(r,g,b)} + 4, & \text{si } b = \max(r, g, b) \end{cases} \quad (1)$$

$$S = \frac{\max(r, g, b) - \min(r, g, b)}{\max(r, g, b)}$$

$$V = \max(r, g, b)$$

De esta manera se efectúa una binarización de las matrices B y V previamente suavizadas con la ecuación (2), para evitar discontinuidades.

$$i'_{x,y} = \frac{1}{k} \sum_{(x,y) \in S(x,y)} i(x,y) \quad (2)$$

Donde k define el tamaño de la ventana empleada para el suavizado de la imagen, i' es el valor del pixel suavizado mediante el filtro promediador y S el conjunto de pixeles en la vecindad del pixel tratado i . Finalmente se aplica la binarización mediante el método automático de Otsu, el cual considera un umbral óptimo cuando la varianza entre clases (3) entrega un valor mínimo.

$$\sigma_w^2(t) = P_1(T)\sigma_1^2(t) + P_2(t)\sigma_2^2(t) \quad (3)$$

Donde P_1 y P_2 son los promedios de clase. Matlab ofrece la función `graythresh()` para efectuar la binarización mediante el método de Otsu a las matrices B y V . Las matrices binarias resultantes son combinadas con una función `and` para detectar los puntos luminosos azules de la imagen. Finalmente es necesario encontrar los objetos circulares que representan el marcador activo. Para efectuar este paso, Matlab permite el etiquetado y la identificación de los objetos encontrados en la imagen mediante las funciones `label()` y `regionprops()` respectivamente. En trabajos posteriores se empleara la segmentación empleando el algoritmo de canny [20] y la transformada de hough [21] para su implementación en otra plataforma computacional.

Una vez que se identifica el círculo correspondiente al marcador, se calcula la coordenada x , y del centroide mediante la ecuación (4).

$$x_m = \frac{\max(x) - \min(x)}{2} \quad y_m = \frac{\max(y) - \min(y)}{2} \quad (4)$$

Finalmente se calcula la distancia euclídea d entre el centroide del marcador localizado en las coordenadas x_m, y_m y el punto deseado x_i, y_i (5)

$$d = \sqrt{(x_i - x_m)^2 + (y_i - y_m)^2} \quad (5)$$

En el modo guiado, si la distancia d es mayor que la tolerancia \square , significa que el paciente aún no ha alcanzado la posición solicitada para satisfacer los puntos que forman la figura patrón, dando paso a la acción correspondiente, es decir el estímulo vibratorio en el marcador. Cuando d es menor o igual a \square se manda un estímulo auditivo al paciente para indicar que se ha llegado al punto deseado, a partir de que se ubica la posición del marcador activo del lápiz óptico, el sistema plotea las coordenadas (x_i, y_i) por las que se desplaza.

El procedimiento se repite hasta terminar la sesión desde la interfaz gráfica. El procedimiento para el método de dibujo libre es muy parecido, salvo que no cuenta con la comparación de los trazos del usuario con un patrón, haciendo una parte de la sesión libre del cálculo del error, y permitiendo al usuario realizar dibujos e iluminarlos.

4. Resultados

Para efectuar las pruebas preliminares se contó con la colaboración de 4 pacientes del sexo masculino entre 12 y 22 años. Se evaluó con este sistema la rehabilitación a los 4 pacientes con problemas de motricidad fina, y 3 voluntarios sanos, permitiendo valorar el incremento en la destreza, velocidad y precisión de las trayectorias patrón

propuestas para este sistema, y comparar los resultados de las pruebas de los pacientes con los voluntarios sanos. Las trayectorias patrón utilizadas en este sistema fueron valoradas por especialistas rehabilitación motriz.

La duración de las sesiones fue de 50 minutos aproximadamente y en función de la disposición de los pacientes se programaron 15 sesiones durante un mes, cada sesión comenzaba con 5 minutos del test de fuerza máxima y mínima, 15 min de dibujo libre y posteriormente 30 min de actividades de seguimiento de patrones, los datos recolectados se encuentran en la tabla 1.

Table 1. Tabla de resultados de las sesiones.

Usuarios	PTF(Sesión 1)	Error %	PTF(Sesión 15)	Error %
Paciente 1	29.5 s	76 %	24.9 s	59 %
Paciente 2	27.8 s	64 %	23.1 s	53 %
Paciente 3	34.5 s	53 %	27.4 s	48 %
Paciente 4	24.3 s	72 %	20.2 s	63 %
Usuario sano 1	14.2 s	45 %	10.9 s	35 %
Usuario sano 2	12.5 s	47 %	9.9 s	33 %
Usuario sano 3	13.6 s	39 %	10.1 s	30 %

Los datos recolectados de los pacientes sometidos a las sesiones de rehabilitación con este equipo demostraron una mejora en el tiempo promedio en que se terminó cada figura (PTF) y una disminución en el porcentaje de error de cada usuario del equipo, tanto de los pacientes como de los usuarios sanos, demostrando un incremento en la coordinación viso-manual de las personas sometidas a las sesiones, que fue el objetivo principal de este equipo.

5. Discusión

El trabajo presentado en este artículo es un equipo auxiliar para la rehabilitación motriz, tanto gruesa (modo libre) como fina (modo guiado), enfocado principalmente a niños entre 6 y 14 años, pero también puede ser utilizado por pacientes adultos, como las pruebas presentadas lo demuestran, con el objetivo ayudar en la recuperación de los usuarios sometidos a este tipo de rehabilitación motriz y a su integración a la sociedad.

Según los resultados de las pruebas realizadas a los pacientes, se detectó un problema para estandarizar el valor de los umbrales de fuerza mínimo y máximo, ya que la variación entre cada paciente era notable, dependiendo tanto del rango de edad como del tipo de lesión, solucionando este problema al incluir un test inicial para determinar el mejor valor de los umbrales mínimo y máximo necesarios para cada paciente, solucionando los problemas de calibración, generalizando su uso a casi cualquier tipo de persona que necesite una rehabilitación de este tipo.

Según la información arrojada por los resultados del sistema, se pudo apreciar que los pacientes sometidos al proceso de rehabilitación con este equipo, presentaron una mejora al cabo de 15 sesiones, como se muestra en la tabla 1, mejorando tanto la velocidad con la que se realizaban las pruebas, así como disminuyendo el porcentaje de error en los ejercicios realizados con el equipo.

Aun no se realizan pruebas con plazos mayores a un mes, es por eso que no es posible aún, medir con certeza el índice de recuperación de los pacientes que auxilian

sus sesiones de rehabilitación motriz con este equipo, sin embargo se esperan hacer más pruebas con pacientes, para recopilar la información necesaria para concluir esta primera etapa del proyecto.

El trabajo futuro, incluirá distintas ergonomías como fundas para el lápiz óptico inalámbrico, acoplándose a las características de los pacientes que utilicen el equipo, así como más actividades de realidad aumentada con algunos elementos interactivos, se pretende poder variar no solo el color, sino también el ancho de las líneas ploteadas, para mejorar la velocidad de iluminación de los dibujos realizados en el modo libre y aumentar la base de datos de figuras patrón.

Agradecimientos. Este trabajo ha sido parcialmente financiado por el Consejo Nacional de Ciencia y Tecnología (CONACYT) y la Dirección General de Educación Superior Tecnológica (DGEST).

Referencias

1. <http://cuentame.inegi.org.mx/poblacion/discapacidad.aspx?tema=P>. Enero 10 2013, 14:00hrs.
2. Andrea, Fernández, et al.: Valoración de discapacidad en parálisis cerebral infantil posterior a tratamiento fisiátrico integral. *MedULA*, vol. 20, pp. 51-60 (2011)
3. B., Johansson: Current trends in stroke rehabilitation. A review with focus on brain plasticity. *Acta Neurol Scand*, vol 123, pp. 147–59 (2011)
4. F.J., Badesa; R., Morales; J. M., Sabater-Navarro; N., Garcia-Aracil; J.M., Azorin; C., Perez: Experiencias en el Desarrollo de un Sistema Robótico para Rehabilitación de Miembro Superior para Pacientes con Daño Cerebral Sobvenido. *Revista Universitaria en Telecomunicaciones, Informática y Control*. Vol. 1, No. 1 (2012)
5. Uri, Feintuch; Liat, Raz; Jane, Hwan;, Naomi, Josman; Noomi, Katz; Rachel, Kizony;, et al.: Integrating Haptic-Tactile Feedback into a Video-Capture-Based Virtual Environment for Rehabilitation. *CYBERPSYCHOLOGY & BEHAVIOR*, Vol. 9, No.2 (2006)
6. Péter, Galambos: Vibrotactile Feedback for Haptics and Telemanipulation: Survey, Concept and Experiment. *Acta Polytechnica Hungarica*, Vol. 9 No. 1 (2012)
7. Juan, Caicedo; Fabio, González; Eduardo, Romero: Prototipo de sistema para almacenamiento y recuperación por contenido en imágenes médicas de histopatología. In *Encuentro de Investigación en Posgrados*. Universidad Nacional de Colombia (2007)
8. T., Broekmans; M., Roelants; G., Alders; P., Feys; H., Thijs; B., Eijnde: Exploring the effects of a 20week whole-body vibration training programme on leg muscle performance and function in persons with multiple sclerosis. *Journal of Rehabilitation Medicine*, Vol. 42, No. 9, pp. 866-872 (2010)
9. Li, Jiang; Juhani, Ruutiainen: Improving finger force control with vibrational haptic feedback for multiple sclerosis. *Proceedings of the IASTED International Conference on Telehealth/Assistive Technologies*, pp. 110-115 (2008)
10. Guadalupe, Salas-López; Oscar, Sandoval-González; Ignacio, Herrera-Aguilar; Albino, Martínez-Sibaja; Otniel, Portillo-Rodríguez; Adriana, Vilchis-González: Design and development of a planar robot for upper extremities rehabilitation with visuo-vibrotactile feedback. Published by Elsevier Ltd, *Procedia Technology* vol. 3, pp. 147-156 (2012)

11. Nick, Mumford; Jonathan, Duckworth; Patrick, Thomas; David, Shum; Gavin, Williams; Peter, Wilson: Upper limb virtual rehabilitation for traumatic brain injury: Initial evaluation of the elements system. *Brain Injury*, vol. 24(5), pp. 780–791 (2010)
12. C., Wall; E., Kentalaa: Effect of displacement, velocity, and combined vibrotactile tilt feedback on postural control of vestibulopathic subjects. *Journal of Vestibular Research*. vol. 20, pp.61-69 (2010)
13. Beom-Chan, Lee1; Jeonghee, Kim; Shu, Chen; Kathleen, Sienko: Cell phone based balance trainer. *Journal of NeuroEngineering and Rehabilitation*, pp. 9-10, (2012)
14. Conrad, Wall III: Application of Vibrotactile Feedback of Body Motion to Improve Rehabilitation in Individuals With Imbalance. *Neurology Section, APTA, JNPT Vol. 34*, pp. 98-104 (2010)
15. Rubén, Posada-Gómez; Oscar, Osvaldo, Sandoval-Gonzalez; Erika, Sánchez-Muñiz: Development of a Motion Analysis System and Human-Machine Interaction through Digital Image Processing and Virtual Reality. *Electrical Communications and Computers (CONIELECOMP)* (2011)
16. Karlin, Bark; Preeya, Khanna; Rikki, Irwin; Pulkit, Kapur; Steven, A.; Jax, Laurel; et al.: Lessons in Using Vibrotactile Feedback to Guide Fast Arm Motions. *World Haptics Conference (WHC), IEEE*, pp. 355 – 360 (2011)
17. Benito, Peñasco-Martín; Ana, de los Reyes-Guzmán; Ángel, Gil-Agudo; et al.: Aplicación de la realidad virtual en los aspectos motores de la neurorrehabilitación. *Rev Neurol*, vol. 51 (8), pp. 481-488 (2010)
18. J., W., Burke: Augmented Reality Games for Upper-Limb Stroke Rehabilitation. *Games and Virtual Worlds for Serious Applications (VS-GAMES), Second International Conference on*, pp. 775-78 (2010)
19. P., gil; F., torres; Detección de objetos por segmentación multinivel combinada de espacios de color. *XXV jornadas de automática, Ciudad Real* (2004)
20. X., Wang; Z., Gongshang; Hangzhou; An Edge Detection Algorithm Based on Improved CANNY Operator; *Intelligent Systems Design and Applications, ISDA, Seventh International Conference on*, pp. 623- 628 (2007)
21. Maji, S.: Object detection using a max-margin Hough transform. *Computer Vision and Pattern Recognition, CVPR. IEEE Conference on*, pp. 1038- 1045 (2009)

Diseño de un sistema auxiliar para rehabilitación motriz mediante realidad virtual y valoración de la fuerza palmar en deficiencias de motricidad fina

Patricia Ivette Cornelio Martínez, Blanca Estela González Sánchez,
Roberto Antonio Montaña Murillo, Rubén Posada Gómez
y Gerardo Águila Rodríguez

División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Orizaba
Orizaba, Veracruz, México

paty_corneli@hotmail.com, blancags67@hotmail.com,
ingeniero_montano@hotmail.com, pgruben@yahoo.com,
gerardo_aguila03@yahoo.com.mx

Resumen En este trabajo se presenta el diseño y desarrollo de un sistema de rehabilitación motriz fina mediante la interacción con ambientes virtuales, cuya finalidad es capturar de una mejor manera la atención de los pacientes, y mejorar la iniciativa parte del usuario en la realización de las sesiones de ejercicios, contribuyendo con esto en mejorar los índices de progreso en la rehabilitación de las personas afectadas con problemas de discapacidad motriz fina, además este equipo cuenta con un sistema de medición de fuerza palmar, diseñado con sensores de tipo resistivo, que detectan la fuerza en cuatro puntos del arco principal de la mano, mostrando y almacenando en una interface gráfica los resultados obtenidos, basándose en esto, se pudo demostrar una mejora en los pacientes sometidos a la rehabilitación con este sistema.

Palabras clave: rehabilitación, motricidad fina, fuerza, interface, realidad virtual.

1. Introducción

En la actualidad, la discapacidad es un problema muy fuerte al nivel mundial con el que vive la sociedad, que impide al afectado tener una buena calidad de vida y desarrollarse normalmente [1]. En México existen 5 739 270 personas que padecen algún tipo de discapacidad, lo que representa 5.1% de la población total, de ésta cifra el 58.3% presenta discapacidad de tipo motriz, es decir, limitaciones para mover las extremidades del cuerpo, esto nos indica la importancia de realizar investigaciones que contribuyan a la rehabilitación de este padecimiento, teniendo en cuenta que la discapacidad motriz representa el problema más fuerte en cuanto a discapacidad en México [2]. Este trabajo está enfocado en la discapacidad motriz fina, es decir, la limitación de los movimientos que implican alta precisión en el uso de la mano, como por ejemplo escribir. Una variable importante que intervine en la rehabilitación

motriz fina es la medición de la fuerza, ya que las personas que presentan ésta discapacidad carecen de la fuerza suficiente para realizar actividades cotidianas, y es por eso que se hacen necesarios dispositivos auxiliares que midan esta variable y demuestren una diferencia en el incremento de la fuerza entre sesiones de rehabilitación. En el desarrollo de equipos que apoyan a la rehabilitación de la discapacidad motriz fina, existen varios que utilizan realidad virtual y video juegos para hacer éste proceso más llamativo para los pacientes, éstos dispositivos fortalecen la coordinación visio manual haciendo uso de la neuroplasticidad, a través de ejercicios de posicionamiento y trazos, utilizando estímulos visuales [3, 4, 5] la desventaja de éstos dispositivos es que no miden la fuerza empleada para el movimiento de la mano. En el 2010 un grupo de investigadores desarrolló un dispositivo que mide la fuerza ejercida sobre un escalpelo, ofreciendo precisión de corte de tejido, empleando una fuerza específica y es autónomo [6] éste tipo de equipos son empleados en el área médica para realizar intervenciones quirúrgicas con mayor precisión. También se han desarrollado equipos de medición de fuerza de las extremidades inferiores como por ejemplo dispositivos para el análisis de la marcha que reconocen acciones como correr, saltar y caminar, utilizando sensores de fuerza resistivos, estos equipos tienen aplicación en el área de deportes [7].

Otra aplicación muy utilizada, es en el sector industrial, en el sensado y automatización de procesos, los cuales emplean celdas de carga para detectar la fuerza exacta en un punto ubicado en determinada maquinaria [8, 9]. Hablando de rehabilitación, se han realizado estudios que indican en comportamiento de la mano en diversas situaciones, por ejemplo en el caso de las personas con Osteoartritis [10] o con síndrome de túnel carpiano [11] analizando las diferencias con personas totalmente sanas, esto con la finalidad de proporcionar datos e información para contribuir a las investigaciones de rehabilitación motriz fina.

Por lo anterior en este trabajo se diseñó un sistema de rehabilitación haciendo uso de ambientes virtuales, que conjunta las técnicas de medición de fuerza, pero, enfocado a la rehabilitación motriz fina como equipo auxiliar, analizando información de estudios anteriores [12, 13], para medir la fuerza en cuatro puntos del arco principal de la mano, utilizando cuatro sensores de fuerza de tipo resistivo y desplegando en tiempo real mediciones en kilogramos y en newtons, demostrando el comportamiento y distribución de las fuerzas en este arco principal, mediante una interface visual que permite la utilización y almacenamiento de los datos mediante formatos digitales, con la finalidad de comprobar un incremento en la fuerza de la mano, y así determinar una rehabilitación motriz fina eficiente en personas con este tipo de discapacidad.

El contenido técnico de este trabajo, presenta en primer lugar la caracterización de los sensores de fuerza, el desarrollo de hardware que incluye el sistema físico empleado, y el desarrollo de software que contiene la interface gráfica de visualización para la modalidad evaluación y las actividades de realidad virtual para la modalidad rehabilitación, al hacer uso de estas herramientas para tratar a cinco personas con problemas de motricidad fina, se obtuvo una mejora en los resultados arrojados en la evaluación.

2. Desarrollo técnico

En esta sección se describe la metodología utilizada para el diseño del sistema de sensado de fuerza, empezando por la caracterización de los sensores, el desarrollo del hardware y software utilizado para la interface de realidad virtual y de resultados. Este trabajo está dividido en cinco módulos, el primero es el circuito de sensado, el segundo módulo corresponde a la tarjeta de adquisición de datos y comunicación, el tercero se trata de una interface de rehabilitación en el cual se realizan ejercicios para fortalecer los músculos de la mano y el antebrazo haciendo uso de juegos de realidad virtual, el cuarto módulo es el de valoración en el cual se muestra la información detectada por los sensores con respecto a la fuerza empleada para realizar los ejercicios, y por último el módulo de visualización de resultados, en la cual se puede apreciar los resultados obtenidos a lo largo del proceso de rehabilitación almacenados en una base de datos para poder apreciar los resultados de cada paciente de manera individual. En la Fig. 1 se muestra los módulos que componen este trabajo.

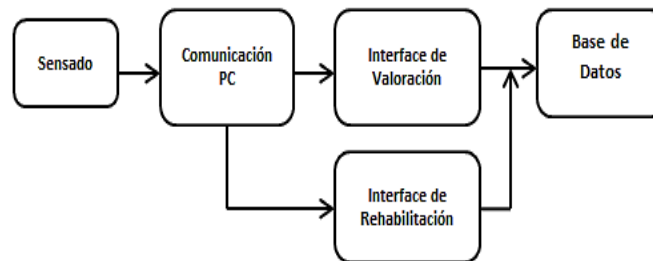


Fig. 1. Diagrama a bloques de los módulos de este trabajo

2.1. Caracterización del sensor

Se utilizó un sensor de fuerza de tipo resistivo (FSR 402) de la compañía Interlink Electronics, el cual varía el valor de su resistencia al ejercer una fuerza sobre él, el rango de medición del sensor es de 10g-10kg y la variación resistiva es de 100Ω-100kΩ [14]. La primera etapa para poder emplear este sensor, fué la de caracterización, en la cual se compararon pesos conocidos con la resultante grafica del sensor, proporcionada en la hoja de datos del fabricante, la unidad de medida de peso que se utilizó en este proceso fué kilogramos, no obstante se realizaron las equivalencias para poder contar con los datos de fuerza en Newtons. En la Fig. 2 se muestra la imagen del sensor FSR y su gráfica de fuerza-resistencia. Según el comportamiento del sensor, mientras mayor sea la fuerza ejercida sobre él, menor será la resistencia. En la Fig. 3 se muestra el circuito propuesto en este trabajo, que consta de un divisor de tensión entre el sensor FSR y una resistencia fija de 10kΩ, llamada R_m , así, al disminuir la resistencia del sensor, el flujo de corriente será mayor y provocará un voltaje incremental en la resistencia R_m , dando como resultado la

ecuación (1). La tarjeta utilizada para la adquisición de los datos es la tarjeta “Arduino Uno R3”, la cual posee 6 canales conectados a un conversor analógico digital de 10 bits. Esto significa que convertirá voltajes entre 0 y 5 volts a un número entero entre 0 y 1023. Esto proporciona una resolución en la lectura de: 5 volts / 1024 unidades, es decir, 0.0049 volts (4.9 mV) por unidad.

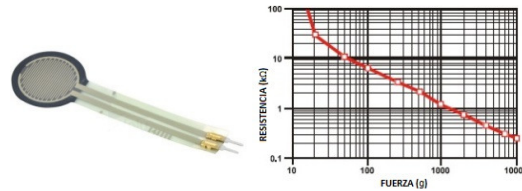


Fig. 2. Sensor FSR y su comportamiento.

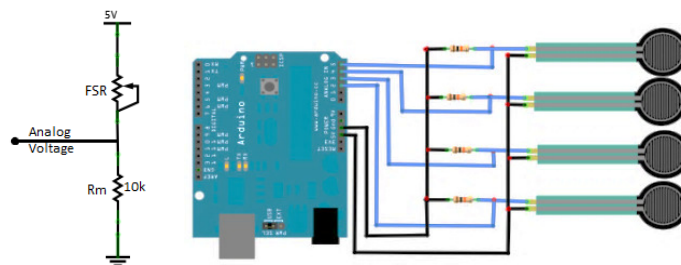


Fig. 3. Diagrama de polarización del FSR y la configuración con los cuatro sensores empleando la tarjeta arduino uno.

$$V_o = V_{cc} \left(\frac{R_m}{R_m + FSR} \right) \quad (1)$$

Para la medición de la fuerza se utilizaron los cuatro sensores resistivos para poder determinar la fuerza del agarre de pinza (arco principal de la mano) de pacientes con problemas de motricidad fina, por ende, se utilizaron cuatro configuraciones similares, una para cada sensor y las salidas de cada divisor de voltaje fueron leídas por los puertos analógicos de la tarjeta de adquisición de datos. Dicho esto, los datos leídos por los puertos analógicos (A0, A1, A2, A3), son registrados y enviados hacia la interface de realidad virtual diseñada en XVR al mismo tiempo que a la interface de visualización, diseñada en visual C#, donde se interpretan estos valores y calculan las conversiones a unidades de fuerza, resistencia y voltaje, representando también en esta interface de manera gráfica, la variación en la fuerza aplicado a cada sensor, mediante cambios cromáticos en zonas preestablecidas, y desplegando las medidas resultantes de las conversiones realizadas.

Tabla 1. Tabla de equivalencia de fuerzas y resistencias

Fuerza (Kg)	Fuerza (N)	FSR (Ω)	Voltaje R (V)	Valor ADC
0	0	inf	0	1024
2	19.61	925	4.54	80
4	39.22	759	4.63	60
6	58.83	583	4.72	40
8	78.45	416	4.8	20
10	98.06	250	4.9	5

En la tabla 1, se muestran las presiones expresadas tanto en kilogramos, como su equivalente en newtons, así como la resistencia del sensor FSR, la corriente y el voltaje de salida del divisor de tensión, los datos se obtuvieron mediante los cálculos realizados en la interface gráfica, por medio de las ecuaciones de conversión en la etapa de procesamiento de las señales de entrada.

2.2. Desarrollo de hardware

La fuerza de agarre es la fuerza aplicada con la mano para apretar o sujetar objetos, la mano tiene una amplia anatomía en cuanto a formas y presiones, este trabajo se enfocó al agarre de pinza del el arco principal, que va desde el dedo pulgar hasta el dedo índice, esto se debe a que éste arco, es el más utilizado para realizar movimientos en los que interviene la motricidad fina. Algunos ejemplos de motricidad fina son: sostener un lápiz y escribir. En la Fig. 4 se muestra la imagen del arco principal de la mano y su funcionalidad, así como los puntos a sensar.

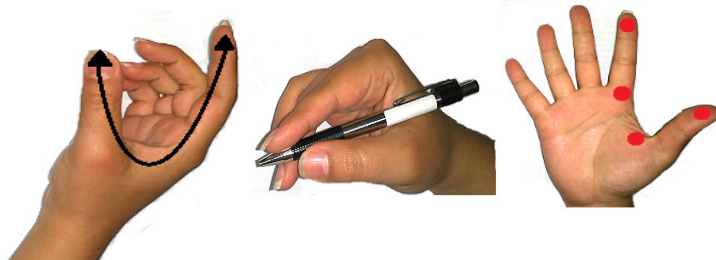


Fig. 4. Arco principal de la mano, agarre de pinza en un ejercicio de motricidad fina y los cuatro puntos a sensar.

Tomando como base la anatomía mostrada en la Fig 4, se realizaron mediciones colocando cuatro sensores distribuidos en un cuerpo cilíndrico de 2.5cm de diámetro y 15cm de altura. Los sensores están dispuestos en la superficie del cilindro a 2cm de la parte superior, posicionados para sensar la presión ejercida por el dedo pulgar, índice y dos puntos intermedios. Con el arreglo anterior se realizaron ejercicios de tipo palmar cilíndrico, los cuales consistieron en ejercer presión con la mano sobre el

cilindro en repetidas ocasiones durante lapsos cortos de tiempo con la intención de generar y capturar señales asociadas con la fuerza del arco principal (Fig. 5).

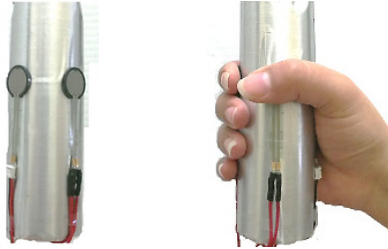


Fig. 5. Cilindro de medición de agarre palmar cilíndrico.

2.3. Desarrollo de software

Este proyecto cuenta con dos modos principales de uso, el primero es como evaluador del índice de rehabilitación de un paciente en función de la valoración de la fuerza palmar cilíndrica, el segundo modo es una serie de ejercicios mediante sesiones en ambientes virtuales que llaman la atención del paciente y favorecen la iniciativa en la realización de actividades que integran la rehabilitación. Hay que tener en mente que este es un sistema auxiliar al proceso de la rehabilitación motriz, y una herramienta de valoración de la fuerza palmar, ya que es muy común que los pacientes con problemas de motricidad fina, cuenten con problemas de tonicidad en los músculos cortos del brazo, o problemas en la coordinación viso-manual, que repercutan en la fuerza y destreza de la mano, y es por esto que este sistema conjunta herramientas de sensado de fuerza palmar, con ejercicios activos y desarrollo de actividades en ambientes virtuales para complementar un sistema que auxilie pacientes con estos problemas.

2.4. Modo de evaluación

Para el primer modo, se necesitaron procesar las señales provenientes de los sensores de fuerza y se diseñó un software mediante “Microsoft Visual Studio C# 2010”, y consta de 4 ventanas (comunicación, datos del paciente, procesamiento, despliegue y graficación de datos), la comunicación de sistema se basa en el protocolo serial, mediante puertos “COM” (USB) configurables desde la primera pantalla de la interface (Fig. 6 izquierda), en esta pantalla también se configura la velocidad de comunicación a la que se desea trabajar con la tarjeta utilizada para la adquisición de datos, una vez establecidos los parámetros de configuración de la comunicación, se continúa a la segunda pantalla (Fig. 6 derecha) donde se introduce el nombre del paciente y otros datos personales para poder llevar un registro de las evaluaciones posteriores a las sesiones de rehabilitación, y con esto medir la eficiencia de la rehabilitación en un paciente en particular o en cada paciente que se someta a tratamiento con este equipo.

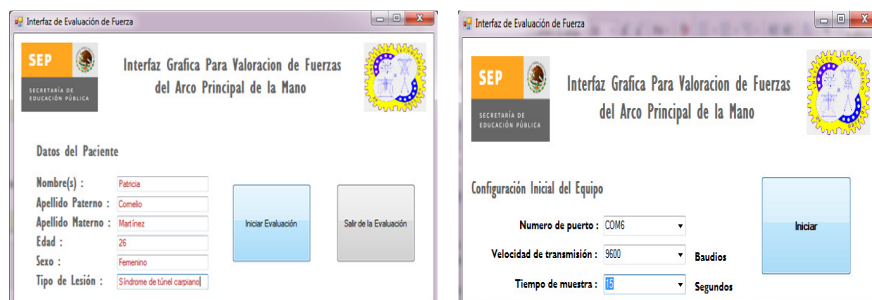


Fig. 6. Ventana de comunicación y ventana de datos iniciales.

Ahora se procede a la captura de los datos para su procesamiento, mediante la tercer pantalla de la interface (Fig. 7 izquierda), donde se muestran las fuerzas aplicadas a los sensores tanto en Kilogramos como en Newtons, así como el promedio de estas fuerzas mostradas en ambas unidades a lo largo de la duración del test (tiempo establecido por el terapeuta), convirtiendo estas fuerzas en variaciones de intensidad de color, es decir, mientras más fuerza se detecte, más intenso se aprecia el color, para mostrar de una manera más amigable las diferencias entre las presiones ejercidas a cada sensor FSR, tanto para el paciente como para el terapeuta, en la tercera pantalla de la interfaz (Fig. 7 derecha) se muestra el desplegado final de la información capturada en la pantalla anterior, así como el porcentaje comparativo de la fuerza en cada uno de los puntos en el arco principal de la mano, junto con la gráfica de los datos de la sesión, también en esta pantalla se puede navegar en la base de datos del sistema, accediendo a los datos de los usuarios registrados en el equipo.



Fig. 7. Ventana de sensado.

El trabajo que desarrolla la interface gráfica, inicia por la adquisición de los datos enviados por la tarjeta de adquisición, y convirtiendo los datos recibidos de 0 a 1024 en su valor correspondiente de voltaje según la tabla 1, y su equivalencia en kilogramos establecida por la comparación de los pesos conocidos registrados en la misma tabla y realizando la conversión a newton mediante las equivalencias de las ecuaciones 2, 3. Donde N son newtons, kgm son kilogramos masa y Kgf son

kilogramos fuerza, resultantes de multiplicar kilogramos masa por a la gravedad como se muestra en la fórmula 2, teniendo como producto Kg m/s^2 , siendo ésta unidad, la equivalente del newton.

$$1\text{Kgf} = (1 \text{kgm})(9.81 \text{ m/s}^2) \quad (2)$$

$$1\text{Kgf} = 9.81 \text{ N} \quad (3)$$

También se realiza un cálculo para determinar el porcentaje de presión detectado por cada sensor, siendo el sensor 1 correspondiente al dedo pulgar, el sensor 4 el correspondiente al dedo índice y los sensores 2 y 3 a los puntos intermedios, dicha información se presenta en la ventana de resultados, para poder establecer el porcentaje de la fuerza que corresponde a cada punto del arco principal, evaluado por este equipo.

2.5. Modo rehabilitación

El segundo modo de uso es el de rehabilitación, y es donde se desarrollan las dinámicas de la sesión de rehabilitación, mediante herramientas de desarrollo de ambientes virtuales que contribuyen a la mejora continua del paciente, este modo cuenta con dos ambientes para interactuar, el primero es la dinámica de inflar globos mediante la presión ejercida sobre el cilindro de medición, en función de la cantidad de fuerza ejercida sobre el cilindro y del número de repeticiones, teniendo como objetivo el reventar los globos, variando la resistencia del globo dependiendo del color del mismo, y seleccionando esto dependiendo del tipo de lesión, la valoración médica previa, y la evolución del paciente en los registros del equipo. En la Fig. 8 se muestra el diseño preliminar de la mano y el cilindro de medición.

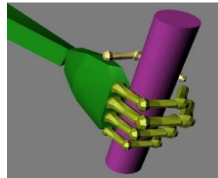


Fig. 8. Estructura de la mano virtual.

El primer ambiente virtual diseñado y asignado como la primera actividad es el inflar globos mediante una perilla como se muestra en la Fig. 9, dependiendo de la presión que se ejerza sobre el cilindro de medición se determinara la velocidad con la que se infla el globo y cada vez que se presione la perilla, el promedio de la fuerza ejercida se suma al promedio de fuerza anterior, de tal manera que cada vez que se presione el cilindro, el promedio de la fuerza registrada será acumulativo hasta cumplir la meta, teniendo como objetivo reventar el globo antes que lo haga un segundo globo controlado por el sistema, dependiendo del nivel en el que se encuentre el paciente se selecciona el color del globo, cada color necesita una sumatoria de fuerza específica, así el primer globo (amarillo) necesita una sumatoria

de fuerza de 25kgm, el segundo globo (verde) requiere una sumatoria de fuerza de 35kgm, el tercer globo (rojo) la sumatoria de fuerza necesaria es de 50kgm y finalmente el cuarto y último globo (azul) requiere de una sumatoria de 70kgm para ser detonado.

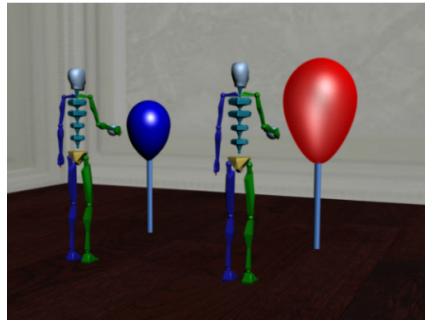


Fig. 9. Sesión de ejercicios con realidad virtual, actividad 1.

El segundo ambiente virtual con el que se desarrollara la siguiente actividad enfocada a la rehabilitación motriz, es el recorrido de un auto a lo largo de una galería de arte, en la que el desplazamiento así como la velocidad del mismo están regidos por la presión del cilindro de medición. En la Fig. 9 (izquierda) se muestra el escenario del pasillo de la galería, por la que se realiza el recorrido, cabe mencionar que al final de los ejercicios, los datos de las mediciones de fuerza son almacenados en la base datos del equipo, para su análisis posterior por un especialista. En la Fig. 9 (derecha) se muestra una de las sesiones realizadas con el sistema en el recorrido del pasillo.

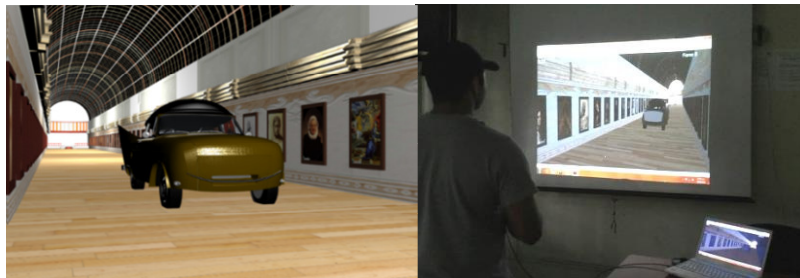


Fig. 10. Actividad 2, paseo en el museo.

3. Resultados

Se obtuvo como resultado un sistema funcional para la medición de la fuerza aplicada a 4 puntos del arco principal de la mano, este equipo, funciona como auxiliar en el proceso de rehabilitación motriz, mediante actividades de realidad virtual y medición de la máxima cantidad de fuerza que un paciente pueda ejercer sobre el

cilindro de sensado, obteniendo un promedio de la fuerza palmar del paciente en el momento del test y comparándola con promedios obtenidos en sesiones anteriores, con esta información, este sistema se utilizó como medio de evaluación, de la variación de fuerza en los pacientes sometidos a terapias de rehabilitación. Con la información obtenida por el sistema y almacenada en la base de datos, es posible cuantificar el índice de incremento en la fuerza máxima de los pacientes sujetos a rehabilitación palmar, y comparar los incrementos obtenidos por diferentes grupos de ejercicios utilizados en sesiones de rehabilitación, haciendo posible, la asignación de ejercicios que presenten mejores resultados para cada paciente. En cuanto a las sesiones de rehabilitación se obtuvieron los siguientes datos (Fig. 10).

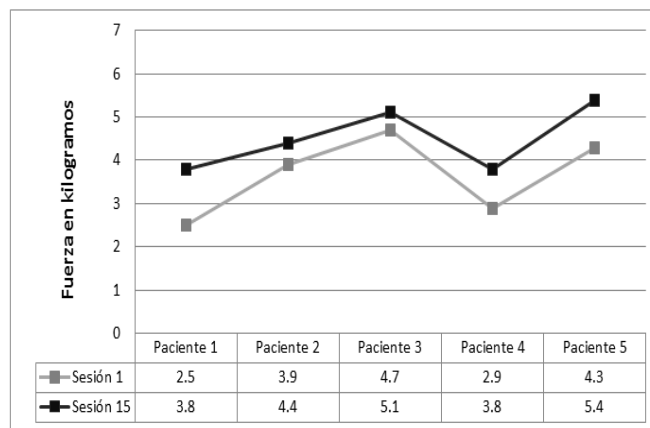


Fig. 11. Graficación de los resultados obtenidos de la primera y última sesión de rehabilitación.

4. Conclusiones

Los resultados obtenidos con el equipo demuestran la eficiencia de las herramientas de realidad virtual integradas al sistema de valoración de la fuerza en pacientes con discapacidad motriz fina, reflejando un incremento en la fuerza de los pacientes que utilizaron el equipo, permitiendo con los datos obtenidos, un más certero diagnóstico por parte de los especialistas para cada paciente y permitiendo enfocarse en terapias que presenten mejores resultados.

Referencias

1. Organización Mundial de la Salud; Informe mundial sobre la discapacidad, 2011.
2. INEGI; Estadísticas a propósito del día del niño; Fondo de Naciones Unidas para la Infancia (UNICEF), 2010.

3. Marlene, Sandlund; Kei, Hoshi; Eva, Lindh, Waterworth; Charlotte, Ha^uger-Ross: A conceptual framework for design of interactive computer play in rehabilitation of children with sensorimotor disorders. *Physical Therapy Reviews* Vol. 14. 2009
4. Mumford, N; Duckworth, J; Thomas, PR.: Shum D, Williams G; Wilson PH.: Upper limb virtual rehabilitation for traumatic brain injury: Initial evaluation of the elements system. *Brain Injury*, 780–791, 2010.
5. Jurgen, Broeren; Katharina, S.; Sunnerhagen, Rydmark; Martin, Mednet: Instituto de Biomedicina de la Universidad " Ir Göteborg: Haptic virtual rehabilitation in stroke: transferring research into clinical practice, *Physical Therapy Reviews*, Vol 14, 2009.
6. Edilberto, Mejía; Jorge, Mejía; Jaime, Delgado; Oscar, Avilés; Darío, Amaya.: Dispositivo transductor para la medición de fuerza y velocidad en una tarea de corte con escalpelo. *Revista EIA*, ISSN,794-1237 p. 147-160, 2010.
7. Darwin, Gouwanda; S. M. N. Arosha Senanayake; M. M. Danushka Ranjana; Marasinghe, Mervin Chandrapal; Jeya Mithra; Kumar, Tung; Mun Hon; Yulius.: Real Time Multi-Sensory Force Sensing Mat for Sports Biomechanics and Human Gait Analysis. *Proceedings of World Academy of Science, Engineering and Technology* vol 29 p.149-154, 2008.
8. Gustavo M. Flores; Rogelio L. Heckera; Esteban Dalhaya; Julián Guzmán.: Diseño preliminar de una celda de carga para maquinado". *Mecánica Computacional*, Vol 25, p. 1347-1355, 2006.
9. L. A. Valdivia Pérez; F. Robles Proenza; R. Santana Milan.: Troquel sensorizado para el estudio de las principales fuerzas que intervienen en el proceso de embutición. *Ingeniería Mecánica*, 37 – 44, 2006.
10. Paula Martins Nunes; Diana Guimarães de Oliveira; Alexander S. Aruin; Marcio José dos Santos.: Relationship between hand function and grip force control in women with hand osteoarthritis *JRRD*, Vol 49, 2012.
11. Wei Zhang; Jamie A. Johnston; Mark A. Ross; Anthony A. Smit; Brandon J. Coakley; Elizabeth A. Gleason; Amylou C. Dueck; Marco Santello.: Effects of Carpal Tunnel Syndrome on Adaptation of Multi-Digit Forces to Object Weight for Whole-Hand Manipulation, *PLoS*, Vol 6, 2011.
12. Pamela Escalona; Jeannette Naranjo; Verónica Lagos; Fresia Solís F.: Parámetros de Normalidad en Fuerzas de Presión de Mano en Sujetos de Ambos Sexos de 7 a 17 Años de Edad, *Rev Chil Pediatr*, 435-443, 2009.
13. Gerwin Smit; Raoul M. Bongers; Corry K. Van der Sluis; Dick H. Plettenburg.: Efficiency of voluntary opening hand and hook prosthetic devices: 24 years of development? *JRRD* Vol 49, p 523–53, 2012.
14. Interlink Electronics - Sensor Technologies, Force Sensor Resistive (FSR 402) data sheet.

Construcción de un sistema de teclado virtual como apoyo pedagógico para personas con habilidades motrices limitadas

Eduardo Emmanuel Ochoa Gracia, José Federico Ramírez Cruz,
Miquelina Sánchez Pulido y Blanca E. Pedroza Méndez

Departamento de Sistemas y Computación,
Instituto Tecnológico de Apizaco, México

e_8a@hotmail.es, federico_ramirez@yahoo.com.mx

Resumen En este artículo se describe el desarrollo de un sistema que hace uso de técnicas de visión por computadora e interacción humano computadora que permita a usuarios con dificultades motrices interactuar con computadoras. El sistema está pensado para funcionar de manera integral con el sistema operativo potenciando así el proceso de enseñanza/aprendizaje en los usuarios mediante el uso de las demás aplicaciones contenidas en el sistema operativo y el acceso a internet. Para lograr la interacción se utiliza un sensor Kinect cuyas características propias facilitan el procesamiento de imágenes del entorno.

Palabras clave: visión por computadora, educación especial, nuevas tecnologías, Kinect, interactividad.

1. Introducción

Con la aparición de nuevas tecnologías tales como la realidad aumentada o la realidad virtual, nuevas oportunidades de aplicación de éstas aparecen como se ve en [1]. La educación es uno de los campos que más puede beneficiarse de estas nuevas tecnologías [2] y actualmente se pueden ver numerosos proyectos alrededor del mundo que se enfocan en la introducción de nuevos sistemas de enseñanza-aprendizaje; sin embargo, existe un área de la educación que carece de aplicaciones para estas nuevas tecnologías, la Educación Especial. Incluso, existiendo muchos sistemas enfocados a la rehabilitación como el propuesto por Da Gama y colaboradores [3], existen pocos enfocados a propósitos educacionales. Además, hay mucho menos aplicaciones enfocadas al aspecto social de la educación.

Este trabajo se centra en el desarrollo de un sistema educacional y social para personas con limitaciones motoras que son incapaces de interactuar específicamente con computadoras.

Nuestro objetivo no es solo hacer un sistema de una sola interfaz, sino uno que trabaje con todo el Sistema Operativo de manera que el usuario incapacitado sea

capaz de manipular la mayoría de las aplicaciones disponibles para éste. Todo lo anterior piensa lograrse mediante el aprovechamiento de las características particulares de las cámaras RGB-D (de color y profundidad), en éste caso un sensor Kinect de Microsoft © el cual está compuesto por una cámara RGB, un proyector infrarrojo y un sensor infrarrojo, así como un conjunto de micrófonos y un motor de inclinación para ajustar el ángulo del dispositivo como se muestra en la figura 1.

La función principal del sistema es la de facilitar la interacción usuario-computadora a usuarios con problemas motrices mediante un teclado cuyo hardware no esté en contacto directo con el usuario.

En la segunda sección de éste trabajo se describirán los procesos realizados para solucionar el problema presentado. En la subsección 2.1 describiremos el proceso de calibración de los sensores del dispositivo Kinect pues existe cierta disparidad entre las imágenes obtenidas por los sensores IR y RGB dados los ángulos de visión de ambos, en la subsección 2.2 se tratará el proceso de solución del seguimiento de la mano mediante el procesamiento de los datos de profundidad y en la subsección 2.3 se hablará de la identificación del teclado mediante el procesamiento de la imagen RGB y el uso de un algoritmo de filtrado de colores. En la tercera sección se muestran los resultados obtenidos y finalmente en la cuarta sección se darán a conocer las conclusiones a las que se ha llegado hasta éste punto del desarrollo.



Fig. 1. Dispositivo Kinect.

2. Solución propuesta

Dado que el sistema busca hacer las funciones de un teclado, la solución del problema se dividió en tres etapas: 1.- Seguimiento de las manos, 2.- Identificación del teclado físico y reconocimiento de las teclas y 3.- Integración con el sistema operativo. En éste artículo se tratarán únicamente las primeras dos etapas.

Para poder realizar cualquiera de las dos funciones antes mencionadas el sensor debe estar enfocando tanto las manos del usuario como el teclado sin que existiera ningún obstáculo entre el dispositivo y estos últimos, por tanto se definió un umbral de visión para el sensor de profundidad que permitiera al sistema ver únicamente los objetos en movimiento (fig. 2.) eliminando el teclado ya que éste será identificado en

la imagen RGB. Dadas las características del sensor de profundidad, éste debe estar situado a más de 40cm de distancia del objeto a seguir de manera que éste sea visible. Una vez establecido el umbral de visión del sensor infrarrojo se debe proceder a realizar el proceso de calibración del sensor IR con el fin de empatar tanto las imágenes RGB como las obtenidas de los datos de profundidad; dicho proceso se detalla en la subsección 2.1.

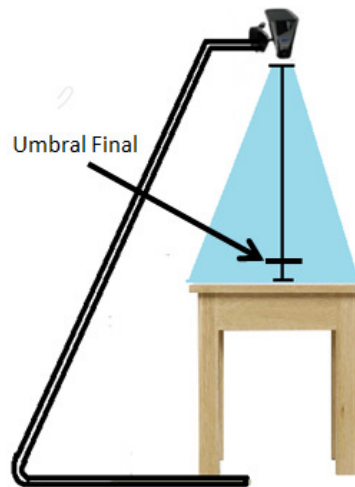


Fig. 2. Posición y distancia del dispositivo y umbral de visión final para el sensor infrarrojo.

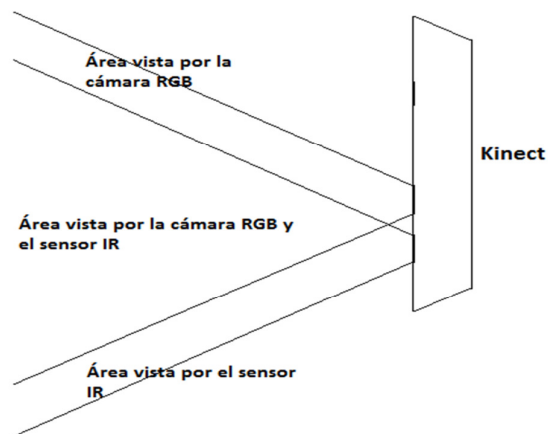


Fig. 3. Disparidad entre área de visible de la cámara RGB y el área visible del sensor infrarrojo.

2.1. Calibración

Con el fin de que los objetos tanto en la imagen de color como en la de profundidad coincidan, fue necesario realizar un proceso de calibrado propio del dispositivo Kinect dado que el sensor infrarrojo y la cámara de color captan distintos ángulos de la misma imagen como lo muestra la figura 3.

Este proceso de calibrado es crucial para el correcto funcionamiento del sistema pues si no se lleva a cabo la posición de la mano sobre el teclado no será obtenida correctamente y ocasionaría problemas en la interacción con el usuario (Esto podría considerarse como una versión simplificada del problema presentado en [4]).

Para solucionar éste problema se hace uso del método `GetAlternativeViewPointCap()` de la función `xn::DepthGenerator` contenido en la librería OPENNI que permite realizar la calibración del sensor IR en relación al sensor RGB.

Ya calibrado el sensor IR es posible iniciar el desarrollo del algoritmo de seguimiento de las manos del usuario como se describe en la subsección siguiente.

2.2. Identificación y seguimiento de la mano

En este punto se llegó a la conclusión de que lo mejor era evitar el proceso de identificación de la mano dadas las características cambiantes de los casos de discapacidad motriz, que va desde problemas de motricidad fina hasta problemas más graves que podrían influir en la forma y posición de la mano y dedos, incluyendo la posibilidad de que el usuario necesite de alguna herramienta para poder interactuar con el teclado como una prótesis, cuya forma no corresponda con la de la mano humana, por lo que el proceso de identificación podría presentar dificultades para encontrar los objetos a seguir.

Para la identificación de los objetos a seguir se definió un umbral de distancia de visibilidad de los objetos en la imagen y en base a éste se realizó la segmentación de los objetos, eliminando todo aquello que estuviese más allá de la distancia definida con el umbral de visibilidad a fin de evitar interferencias en la imagen. Una vez teniendo los objetos segmentados en la imagen éstos son etiquetados para así poder trabajar cada uno por separado; después, se crea una envoltura convexa para cada objeto con el fin de reducir sus características y encontrar el centro geométrico (centroide) de cada objeto [5] mediante la aplicación de la ecuación (1), en donde x y y son las coordenadas de cada pixel i en la región de la imagen en que se trabaja; mientras que k corresponde al número de pixeles contenidos en la región. El centroide de cada región se utiliza como punto de referencia para eliminar el resto del brazo en la imagen dejando sólo el área de la imagen correspondiente a la mano o, en su defecto, al extremo del objeto a seguir (ver la figura 4).

$$\bar{x} = \frac{\sum_{i=0}^k x_i}{k}, \bar{y} = \frac{\sum_{i=0}^k y_i}{k} \quad (1)$$

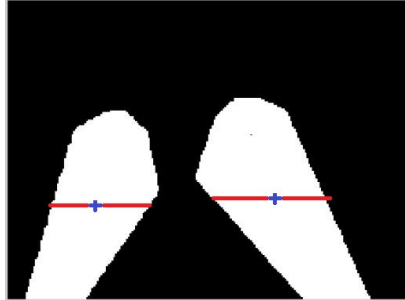


Fig. 4. Primeros centroides encontrados en cada región segmentada, utilizados para recortar las imágenes.

Después de este proceso se obtiene por segunda vez el centroide de las regiones obtenidas tras reducir el área a seguir como se muestra en la figura 5.

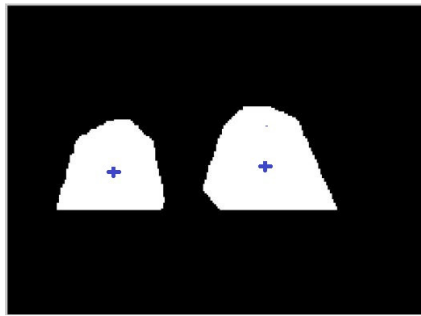


Fig. 5. Segundos centroides encontrados en las imágenes recortadas y utilizados como puntos a seguir en la imagen.

Al ser éste proceso realizado independientemente para cada región en la imagen logramos seguir individualmente cada objeto y no interfieren el uno con el otro a menos que los objetos se unan en cierto momento o que alguno desaparezca del cuadro, en cuyo caso sólo se seguirá el objeto restante en la imagen.

Una vez solucionados los problemas de la calibración del sensor IR y del seguimiento del usuario es posible adentrarse en el desarrollo del algoritmo de identificación del teclado y su integración con el seguimiento del usuario como se detalla en la subsección 2.3.

2.3. Identificación del teclado físico

En el caso del teclado se necesitan definir características que servirán para reconocerlo, a partir de dichas características se establecen los puntos de anclaje de los que se obtendrán las proporciones para cada tecla, siendo éste método invariable al escalamiento.

Mediante un filtro de color se elimina de la imagen todo aquello que no sea de utilidad dejando únicamente las áreas que cumplen con los requisitos necesarios para permanecer en la imagen como se observa en la figura 6b en cuyo caso se optó por filtrar los tonos rojos, eliminando toda mezcla de colores en las que el color rojo sea menor que el verde y azul, así como cuando la diferencia entre los tonos azules y verdes y los tonos de rojo sea superior a 100 en una escala de 255 dado el esquema de color RGB, dejando visibles únicamente los cuadros a las esquinas del patrón de prueba.

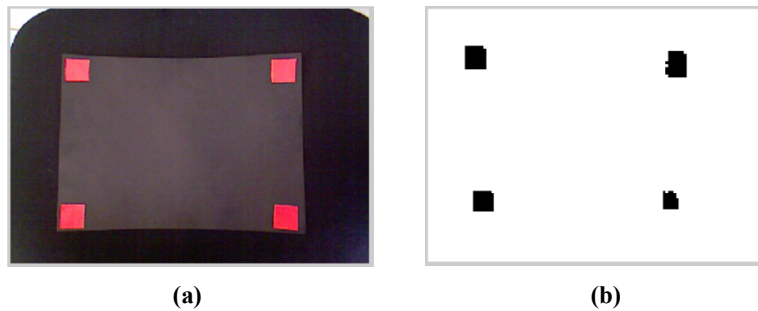


Fig. 6. Prueba de detección de los puntos de anclaje con un filtro de color; (a) Imagen vista por la cámara de color; (b) segmentación de los puntos de referencia para ubicar el teclado.

Después de aplicar el filtro de color para detectar los puntos de anclaje, se etiquetan las áreas segmentadas en la imagen para encontrar el centro de cada una de éstas. Utilizando como referencia un eje cartesiano, se realiza la diferencia de las distancias entre los objetos superiores sobre el eje X (ancho) y otra entre los dos objetos ubicados a la izquierda de la imagen sobre el eje Y (alto). Las distancias obtenidas se dividen entre el número de teclas contenidas en el teclado a lo ancho y a lo alto, obteniendo así las áreas activas para cada tecla (figura 7), las cuales variarán en base a la distancia entre el teclado y la cámara pero sin perder la ubicación de las teclas.

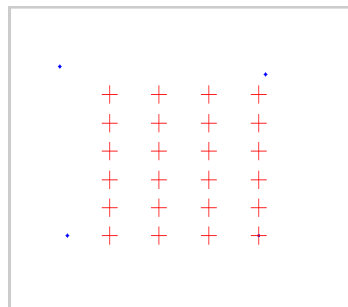


Fig. 7. Áreas activas de las teclas

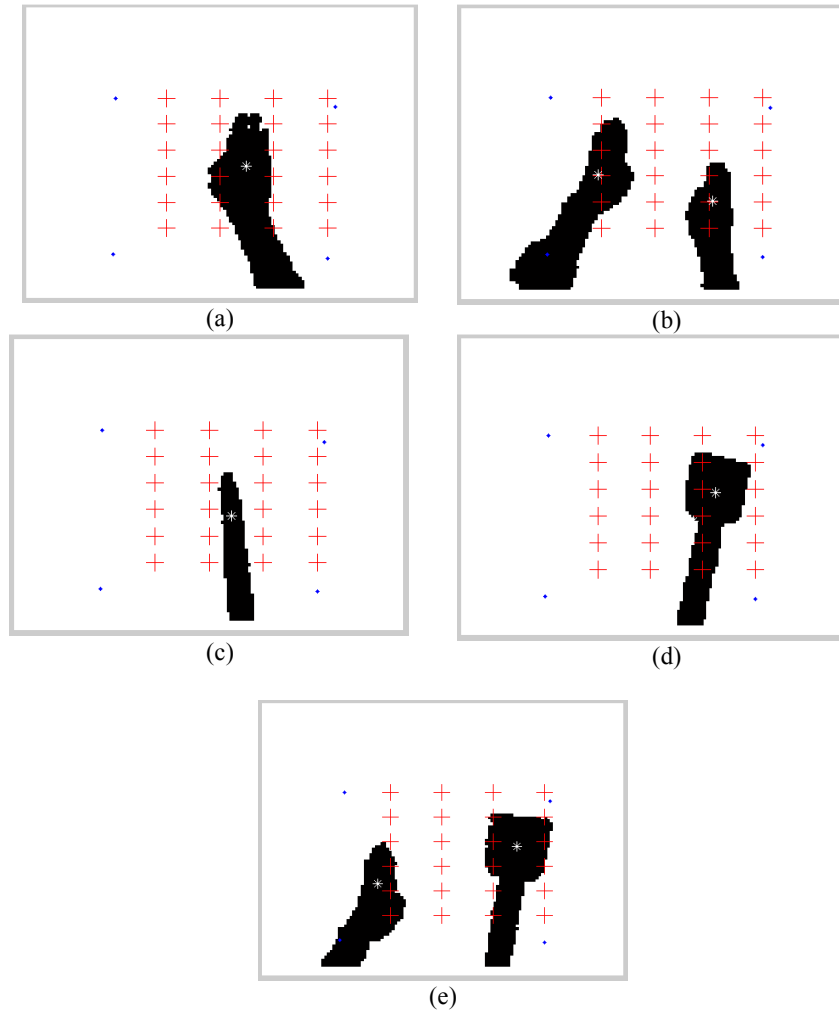


Fig. 8. Resultados del seguimiento con las manos y diversos objetos: (a) Una sola mano; (b) dos manos; (c) Un objeto cilíndrico delgado; (d) Una pala y (e) Una pala y una mano.

3. Resultados preliminares

Para las pruebas se utilizó un patrón rectangular de 100 x 70 cm., con 4 cuadrados rojos en las esquinas de 2x2 cm., un dispositivo Kinect situado a 70 cm. de distancia del patrón de prueba y diversos objetos.

Al aplicar ambos algoritmos, tanto el de seguimiento de objetos como el de detección del teclado e imprimir los objetos de interés en una sola imagen obtenemos los resultados observados en la figura 8. En la figura 8a se muestra la presencia de

una sola mano en negro; se indican los centros de cada tecla con las cruces rojas, y el centroide de la mano con una estrella blanca dentro de la mano. En la figura 8b se muestran dos manos con las mismas características. En la figura 8c se utilizó un utensilio (un objeto cilíndrico delgado) en el cual también se muestran las mismas características que en las figuras anteriores. En la figura 8d se muestra la presencia de una pala y por último en la figura 8e se muestra una mano y la pala.

Como se ve en las serie de imágenes de la figura 8 el sistema detecta cualquier utensilio utilizado para hacer interacción con el teclado virtual y no habrá algún inconveniente por la forma que tengan estas herramientas.

Hasta ahora sólo se ha realizado la detección de las manos o utensilios utilizados para interactuar con el teclado; sin embargo, aún falta realizar el proceso de activación de las señales o comandos necesarios para emular el teclado de la computadora. Para detectar la interacción de la mano con la tecla se utilizará la imagen de profundidad proporcionada por el sensor y se tratará de detectar el movimiento vertical de la mano. Este es un proceso un poco complejo que se está resolviendo.

4. Conclusiones y trabajos futuros

El presente trabajo presentó los avances de la realización de un teclado virtual utilizando una cámara de profundidad y una cámara de color, incluidos en el sensor Kinect de Microsoft. El sistema realizado hasta ahora presenta buen desempeño tanto en el seguimiento de las manos del usuario o utensilios utilizados para interactuar con él, así como en la detección del teclado físico, asignando satisfactoriamente las áreas activas para cada tecla. La segmentación de los objetos se realizó con la cámara de profundidad (IR) y la identificación del teclado con la cámara de color (RGB). El sistema se probó con movimientos en tiempo real sin tener problemas de ralentización y fue implementado utilizando la herramienta Matlab.

En trabajos posteriores se procederá a detectar el movimiento vertical para detectar la interacción física de la mano o el utensilio con el teclado, además de realizar la interacción con el Sistema Operativo de la computadora y realizar las pruebas correspondientes de desempeño.

Referencias

1. DePriest, D.: The Fifth Dimension: How Augmented Reality is Launching Worlds within, TCC Proceedings 2012, pp. 6-13 (2012).
2. Justina Hsu, H.: The potential of Kinect in education. International Journal of Information and Education Technology, pp. 356-370 (2011)
3. Da Gama, A., Chaves, T., Figueiredo, L., and Teichrieb, V.: Poster: Improving motor rehabilitation process through a natural interaction based system using Kinect sensor. 3D User Interfaces (3DUI), 2012 IEEE Symposium, pp. 145-146 (2012)

4. Nooan, Philip J., Hallet, William A.: The Design and Initial Calibration of an Optical Tracking System Using the Microsoft Kinect, 2011 IEEE Nuclear Science Symposium Conference Record, pp. 3614-3617 (2011)
5. Park, H.: A Method for Controlling Mouse Movement using a Real-Time Camera.
6. Barilovits, D., DePriest, K.: LIVE: Xbox Kinect's Virtual Realities to Learning Games, TCC 2011 Proceedings, pp. 48-54 (2011)

Guiado háptico pasivo sub-óptimo con observador de energía

Mario O. Ordaz-Oliver¹, Omar A. Domínguez-Ramírez²
y Omar J. Santos-Sánchez³

Universidad Autónoma del Estado de Hidalgo,
Grupo de Electrónica y Control, CITIS-ICBI, Mineral de la Reforma, Hidalgo, México
x2_kyo@hotmail.com, omar@uaeh.edu.mx, omarj@uaeh.edu.mx

Resumen En este artículo se implementa una estrategia de control no lineal sub-óptima sobre la interfaz háptica PHANToM 1.0 para tareas de guiado háptico, la cual se sintetiza a partir de la teoría de Programación de Dinámica de Richard Bellman. El control implementado en esta contribución mantiene la estructura del enfoque de control sub-óptimo. El objetivo principal de la aplicación de este controlador no lineal sub-óptimo es la penalización de un índice de desempeño basado en la energía total del sistema, que garantice un comportamiento permisible en tareas de guiado háptico, preservando la vida útil de los componentes electromecánicos del dispositivo, y la medición objetiva de desempeño a través del intercambio de energía en la interacción hombre-robot.

Palabras clave: programación dinámica; sub-óptimo; interfaz háptica; estabilidad; pasividad.

1. Introducción

La optimización de sistemas dinámicos es un área recientemente estudiada. La optimización de la energía es un objetivo natural en distintas áreas de la ingeniería, así como el análisis y control de los sistemas dinámicos [1,8]. Se han propuesto distintas técnicas y objetivos para la optimización, sin embargo, hay dos ejes de mayor investigación en esta área: El primero de estos enfoca su interés en la síntesis de estrategias de control óptimo, y el segundo eje concentra su interés en el filtrado óptimo de señales [2]. En el campo de la optimización no lineal hay una serie de esfuerzos por obtener un controlador para los sistemas no lineales [10], que consideran sistemas de parámetros distribuidos. Cuando se concideran los enfoques de programación dinámica y cálculo variacional para determinar la solución de problemas de control óptimo en sistemas no lineales, se utiliza comúnmente la ecuación Halmilton Jacobi Bellman. Sin embargo, se sabe que la función de Bellman para sistemas no lineales no tiene una estructura definida y por tanto su aproximación o determinación son una tarea bastante complicada. Cuando la programación dinámica es aplicada al control de robots manipuladores, la complejidad del problema aumenta, puesto que deben considerarse distintos fenómenos como las zonas singulares, los efectos eléctricos y

mecánicas a los que están constantemente sometidas las articulaciones de los robots manipuladores. Si a esto se agrega el propósito de optimizar el desempeño dinámico del robot por medio de la función de Bellman o su aproximación, la complejidad del problema se incrementa aún más. En esta área, se presentan principalmente aquellos resultados definidos por la planificación de trayectorias con la optimización de la ley de control y el problema de estabilización óptima. Hemami en [4] obtiene una estrategia de control óptimo por medio de la linealización de la dinámica y la obtención de la solución de la ecuación algebraica de Riccati relacionada a tal linealización, con el fin de obtener un controlador para tareas de seguimiento robots móviles, esta propuesta considera un índice de desempeño cuadrático de horizonte infinito. Por otra parte Lin, F. y Brandt en [9], emplean un regulador cuadrático lineal (LQR) para la obtención de una ley de control robusto para un robot tipo SCARA de 2 articulaciones, cuyo modelo matemático es desconocido y se utiliza la ecuación de Hamilton Jacobi Bellman, por medio de la cual se sintetiza un controlador óptimo, con la solución de una ecuación algebraica de Riccati. Verscheure en [14] obtiene la solución al problema de seguimiento de trayectoria en tiempo óptimo para un manipulador industrial KUKA 361 de 6 grados de libertad (GDL), como un problema de control óptimo convexo de una entrada. Es importante mencionar que la literatura examinada y citada sólo presenta resultados de implementaciones numéricas, sin reportar una validación experimental. Hacemos énfasis en la principal contribución de este trabajo, que es poner a prueba un controlador no lineal sub-óptimo en lazo cerrado basado en el enfoque de Programación Dinámica, para guiado háptico, penalizando el índice de desempeño y de este modo obtener un comportamiento apropiado para esta clase de tareas. Esta ley de control fue probada previamente para controlar un péndulo sub-actuado de dos grados de libertad conocido como pendubot [12], el cual presenta un buen desempeño y manipulabilidad. Es importante mencionar que el dispositivo sobre el cual se está probando experimentalmente ley de control no lineal es un sistema robótico completamente actuado. Se presentan los resultados experimentales para validar la viabilidad de este controlador. La determinación de las penalizaciones del controlador sub-óptimo no lineal probado para guiado háptico presenta un grado de complejidad, dado que este control inicialmente se sintetiza con el objeto de minimizar un índice de desempeño. Por otra parte, la penalización en línea del controlador permite la obtención de un comportamiento deseado. El esquema propuesto se prueba con los resultados experimentales mediante el uso de la interfaz háptica PHANTOM 1.0, en una tarea de guiado háptico. Además, utilizando una penalización adecuada del estado y de la ley de control en el índice de desempeño para la síntesis del controlador no lineal, los resultados experimentales obtenidos indican que el algoritmo de control sub-óptimo tiene un desempeño adecuado, el cual puede emplearse con propósitos de rehabilitación, adiestramiento y entrenamiento por medio de tareas de guiado háptico pasivo.

El presente artículo tiene la siguiente estructura: En la sección 2, se describe la plataforma empleada para esta contribución, el modelo matemático y las

propiedades dinámicas de PHANToM 1.0. En la sección 3 se sintetiza la ley de control no lineal sub-óptima de esta contribución. La sección 4 se emplea para describir la arquitectura y el hardware de la interfaz háptica PHANToM 1.0 y a partir de ello se definen las condiciones para la evaluación de la ley de control no lineal. En la Sección 5, se dan a conocer los resultados experimentales de la implementación. Finalmente, en la sección 6 se definen las conclusiones y observaciones finales.

2. Modelo matemático y propiedades del sistema

Inicialmente, en esta sección se presenta el dispositivo háptico PHANToM 1.0, sistema sobre el cual se evalúa la estrategia de control no lineal sub-óptimo. A partir del modelo dinámico del sistema y de sus propiedades, se sintetiza la ley de control a probar. PHANToM 1.0 es un dispositivo háptico de tres grados de libertad, que cuenta con articulaciones de revolución y por tanto puede ser definido como un mecanismo de eslabones articulados, tiene propósitos distintos a los de un robot manipulador, sin embargo, la técnica de modelado es la misma. De acuerdo con [13], el modelo dinámico reducido de este dispositivo es obtenido por medio de la formulación de Euler-Lagrange en movimiento libre, y definido como:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + b\dot{q} + G(q) = \tau, \quad (1)$$

más específicamente un dispositivo háptico en aplicaciones de guiado háptico, el sistema define la formulación Euler-Lagrange (con el operador humano en el lazo de control)

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + b\dot{q} + G(q) = \tau + \tau_h, \quad (2)$$

con $H(q) \in \mathbb{R}^{3 \times 3}$ matriz de inercias, $C(q, \dot{q}) \in \mathbb{R}^{3 \times 3}$ matriz de fuerzas centrífugas y de coriolis, $b \in \mathbb{R}^{3 \times 3}$ matriz de fuerzas de fricción, $G(q) \in \mathbb{R}^{3 \times 1}$ vector de fuerzas debidas a efectos gravitacionales, $[\tau + \tau_h] \in \mathbb{R}^{3 \times 1}$ vector de pares generalizados de entrada (definidos por la ley de control y el operador humano), en donde $\tau_h = J^\top F_h$, y F_h representa la dinámica de fuerza que propicia el operador humano en el efector final, q, \dot{q} y $\ddot{q} \in \mathbb{R}^{3 \times 1}$ vectores de posiciones, velocidades y aceleraciones generalizadas respectivamente, con la forma:

$$H(q) = \begin{bmatrix} h_{11}(q_2, q_3) & 0 & 0 \\ 0 & h_{22} & h_{23}(q_2, q_3) \\ 0 & h_{32}(q_2, q_3) & h_{33} \end{bmatrix},$$

$$C(q, \dot{q}) = \begin{bmatrix} c_{11}(q_2, q_3, \dot{q}_2, \dot{q}_3) & c_{12}(q_2, q_3, \dot{q}_1) & c_{13}(q_2, q_3, \dot{q}_1) \\ c_{21}(q_2, q_3, \dot{q}_1) & 0 & c_{23}(q_2, q_3, \dot{q}_3) \\ c_{31}(q_2, q_3, \dot{q}_1) & c_{32}(q_2, q_3, \dot{q}_3) & 0 \end{bmatrix}, b = \begin{bmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{bmatrix},$$

$$G(q) = \begin{bmatrix} 0 \\ g_2(q_2) \\ g_3(q_3) \end{bmatrix}, \tau + \tau_d = \begin{bmatrix} \tau_1 + \tau_{h1} \\ \tau_2 + \tau_{h2} \\ \tau_3 + \tau_{h3} \end{bmatrix},$$

donde $H(q)$ es simétrica, positiva definida, con elementos que son función de q , como:

$$H(q) = H(q)^T, \\ x^T H(q) x > 0; \forall x \in \mathbb{R}^{n \times 1},$$

$C(q, \dot{q})$ se relaciona con la matriz de inercias $H(q)$ por medio de la expresión:

$$x^T \left\{ \dot{H}(q) - 2C(q, \dot{q}) \right\} x \equiv 0; \forall x \in \mathbb{R}^{n \times 1}, \quad (3)$$

conocida como propiedad de anti-simetría.

Por otra parte, el sistema en la forma de Euler-Lagrange involucra su energía total, la cual se define como:

$$\varepsilon = \mathcal{K} + \mathcal{U}, \quad (4)$$

en donde ε define el Hamiltoniano o energía total del sistema, \mathcal{K} es la energía cinética y \mathcal{U} la energía potencial, definidas como:

$$\mathcal{K} = \frac{1}{2} \sum_{i=1}^n m_i v_i^2 = \frac{1}{2} \dot{q}^T D(q) \dot{q},$$

$$\mathcal{U} = \sum_{i=1}^n m_i h_i g,$$

sí se deriva (4) y se emplea la propiedad de antisimetría (3), se obtiene:

$$\dot{\varepsilon} = \dot{q}^T H(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{H}(q) \dot{q} + \dot{q}^T G(q), \\ \dot{\varepsilon} = \dot{q}^T \{-C(q, \dot{q}) \dot{q} - b\dot{q} - G(q) + \tau + \tau_h\} + \frac{1}{2} \dot{q}^T \dot{H}(q) \dot{q} + \dot{q}^T G(q), \\ \dot{\varepsilon} = \dot{q}^T \tau + \dot{q}^T \tau_h. \quad (5)$$

Ahora bien, se tiene de la propiedad de pasividad como:

$$V(x) - V(x_0) \leq \int_0^t y^T(\tilde{x}) u(\tilde{x}) d\tilde{x},$$

donde $V(x)$ es una función energética, $y(\tilde{x})$ es la salida, y $u(\tilde{x})$ es la entrada del sistema, esta propiedad establece que un sistema expresado en forma Euler-Lagrange es disipativo, es decir, este no puede disipar más energía de la que almacena. Usando (5) en el sistema Euler-Lagrange, la función de energía ε como función energética, de la siguiente manera se tiene:

$$\varepsilon(t) - \varepsilon(0) \leq \int_0^t \dot{q}^T \tau + \dot{q}^T \tau_h dt. \quad (6)$$

3. Síntesis de la estrategia de control no lineal

En esta sección se sintetiza la estrategia de control que se aplicará al dispositivo háptico para tareas de guiado háptico pasivo. Desarrollamos una ley de control no lineal sub-óptima para un dispositivo háptico cuya dinámica resulta en la forma de un sistema no lineal afín. Se presentan también los avances en el control óptimo de sistemas no lineales, utilizando el enfoque de control por medio de funciones de Lyapunov y programación dinámica, aquí omitimos la prueba de estabilidad, puesto que el objetivo principal del presente artículo se centra en los resultados experimentales de esta ley de control penalizada, y aplicada al dispositivo PHANTOM 1.0, para guiado háptico.

La formulación del problema esta dada alrededor del punto de equilibrio estable, x_{eq} , entonces el sistema no lineal en la formulación Euler-Lagrange (1), el modelo matemático puede expresarse como:

$$\frac{d}{dt} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ H(q)^{-1} [\tau + \tau_h - C(q, \dot{q}) \dot{q} - b\dot{q} - G(q)] \end{bmatrix}, \quad (7)$$

Para obtener la forma estándar de un sistema no lineal de las ecuaciones dinámicas del robot, se tiene que:

$$\ddot{q} = \begin{bmatrix} \dot{q} \\ -H^{-1}(q) \{C(q, \dot{q}) \dot{q} + b\dot{q} + G(q)\} \end{bmatrix} + \begin{bmatrix} 0_{n \times n} \\ H^{-1}(q) \tau + H^{-1}(q) \tau_h \end{bmatrix}, \quad (8)$$

proponiendo el cambio de variables: $q_1 = x_1$, $q_2 = x_2$, $q_3 = x_3$, $\dot{q}_1 = x_4$, $\dot{q}_2 = x_5$, $\dot{q}_3 = x_6$ y definiendo $\bar{x}_1 = [x_1 \ x_2 \ x_3]^\top$, $\bar{x}_2 = [x_4 \ x_5 \ x_6]^\top$, $\tau + \tau_d = u$, reescribimos (8) como

$$\dot{\tilde{x}} = \begin{bmatrix} \bar{x}_2 \\ -H^{-1}(\bar{x}_1) \{C(\bar{x}) \bar{x}_2 + b\bar{x}_2 + G(\bar{x}_1)\} \end{bmatrix} + \begin{bmatrix} 0_{n \times n} \\ H^{-1}(\bar{x}_1) u \end{bmatrix} \quad (9)$$

donde

$$H(\bar{x}_1) = \begin{bmatrix} h_{11}(x_2, x_3) & 0 & 0 \\ 0 & h_{22} & h_{23}(x_2, x_3) \\ 0 & h_{32}(x_2, x_3) & h_{33} \end{bmatrix},$$

$$C(\bar{x}) + b = \begin{bmatrix} c_{11}(x_2, x_3, x_5, x_6) + b_{11} & c_{12}(x_2, x_3, x_4) & c_{13}(x_2, x_3, x_4) \\ c_{21}(x_2, x_3, x_4) & b_{22} & c_{23}(x_2, x_3, x_6) \\ c_{31}(x_2, x_3, x_4) & c_{32}(x_2, x_3, x_6) & b_{33} \end{bmatrix},$$

$$G(\bar{x}_1) = \begin{bmatrix} 0 \\ g_2(x_2) \\ g_3(x_3) \end{bmatrix}, u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix},$$

con $\tilde{x} = x - x_{eq}$, el sistema (1) es reescrito como:

$$\dot{\tilde{x}} = f(\tilde{x}) + g(\tilde{x}) u, \quad (10)$$

en donde $\tilde{x} \in \mathbb{R}^{n \times 1}$ es el estado del sistema y $u \in \mathbb{R}^{1 \times p}$ es la entrada de control, con funciones $f(\tilde{x})$ y $g(\tilde{x})$ continuas y en espacios apropiados, definidas como:

$$f(\tilde{x}) = \begin{bmatrix} \tilde{x}_2 \\ -H^{-1}(\tilde{x}_1) \{C(\tilde{x})\tilde{x}_2 + b\tilde{x}_2 + G(\tilde{x}_1)\} \end{bmatrix}, g(\tilde{x}) = \begin{bmatrix} 0_{3 \times 3} \\ H^{-1}(\tilde{x}_1) \end{bmatrix}. \quad (11)$$

Motivados ahora por en hecho de que el sistema (10) puede ser linealizado alrededor de un punto de equilibrio x^+ , y su linealización se expresa como:

$$\dot{\tilde{x}} = A\tilde{x} + Bu, \quad (12)$$

en donde:

$$\left. \frac{\partial f(\tilde{x})}{\partial \tilde{x}} \right|_{x=x^+, u=u^+} = A, \quad \left. \frac{\partial g(\tilde{x})}{\partial u} \right|_{x=x^+, u=u^+} = B. \quad (13)$$

El par (A, B) es controlable y existe una matriz $Q = \mathcal{H}\mathcal{H}^\top$ tal que el par (A, \mathcal{H}) es observable.

Tomando en cuenta que el método de control sub-óptimo de sistemas dinámicos de esta propuesta se basa en el análisis de la energía. Y para el problema de control sub-óptimo, una funcional puede ser minimizada de acuerdo a criterios deseados propuestos. Esta funcional consiste en la ecuación de energía total del sistema y la energía entregada por sus actuadores. Para esto, definimos el índice de desempeño como:

$$J = \int_0^\infty [f_0(\tilde{x}(t), u(t))] dt, \quad (14)$$

donde f_0 es una función estrictamente positiva definida. De la teoría de Programación Dinámica de Richard Bellman, sabemos que sí existe una función definida positiva $V(\tilde{x}^*(t))$, que es continuamente diferenciable y que satisfaga:

$$\left. \frac{dV(\tilde{x}^*(t))}{dt} \right|_{(12)} + f_0(\tilde{x}^*(t), u^*(t)) = 0, \quad (15)$$

entonces u^* es el control óptimo.

Con el fin de proponer un control regulador lineal para el sistema linealizado (12), es posible resolver la ecuación algebraica de Riccati:

$$A^\top P - PBR^{-1}B^\top P + PA = -Q, \quad (16)$$

para obtener la matriz P , donde Q es una matriz simétrica, semi-definida positiva y R es una matriz simétrica, definida positiva. Por medio de la solución P de la ecuación (16) y en base a (14), definimos la función:

$$V(\tilde{x}) = \frac{1}{2}k_E \tilde{\varepsilon}(\tilde{x}_1, \tilde{x}_2)^2 + \frac{1}{2}\tilde{x}^\top \underbrace{\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}}_P \tilde{x}, \quad (17)$$

donde $\tilde{\varepsilon}(\tilde{x}_1, \tilde{x}_2)$ es la función de error de energía, dada como:

$$\tilde{\varepsilon}(\tilde{x}_1, \tilde{x}_2) = \varepsilon(\bar{x}_1, \bar{x}_2) - \varepsilon_d(\bar{x}_1, \bar{x}_2), \quad (18)$$

diferenciando la ecuación (16) a lo largo de las trayectorias de (10) se tiene que:

$$\dot{V}(\tilde{x}) = k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \dot{\tilde{\varepsilon}}(\bar{x}_1, \bar{x}_2) + \tilde{x}^\top \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \dot{\tilde{x}}, \quad (19)$$

dada la dinámica del error, y por las propiedades de las matrices positivas definidas y simétricas, se cumple lo siguiente:

$$\dot{\tilde{\varepsilon}}(q, \dot{q}) = \frac{1}{2} \dot{\tilde{q}}^\top D(q) \ddot{\tilde{q}} + \frac{1}{2} \dot{\tilde{q}}^\top \dot{D}(q) \dot{\tilde{q}} + \frac{1}{2} \ddot{\tilde{q}}^\top D(q) \dot{\tilde{q}} + \dot{\tilde{q}}^\top G(q), \quad (20)$$

por medio de la integral de la propiedad (5), se verifica la propiedad de pasividad (6), como:

$$\int_0^t \dot{\tilde{\varepsilon}}(q, \dot{q}) dt = \int_0^t \dot{\tilde{q}}^\top \tau + \dot{\tilde{q}}^\top \tau_h dt, \Rightarrow \dot{\tilde{\varepsilon}}(q, \dot{q}) - \dot{\tilde{\varepsilon}}(0, 0) = \int_0^t \dot{\tilde{q}}^\top \tau + \dot{\tilde{q}}^\top \tau_h dt, \quad (21)$$

en donde $\dot{\tilde{q}}$ se toma como la salida y τ como la entrada del sistema. Ahora bien, (18) puede escribirse como:

$$\dot{V}(\tilde{x}) = k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \tilde{x}_2 \tau + k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \tilde{x}_2 \tau_h + \tilde{x}^\top \begin{bmatrix} \bar{P}_{11} & \bar{P}_{12} \\ \bar{P}_{21} & \bar{P}_{22} \end{bmatrix} \dot{\tilde{x}} \quad (22)$$

es decir:

$$\dot{V}(\tilde{x}) = k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \tilde{x}^\top \tau + k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \tilde{x}^\top \tau_h + \tilde{x}_1^\top \bar{P}_{11} \tilde{x}_2 + \tilde{x}_2^\top \bar{P}_{21} \tilde{x}_2 + \tilde{x}_1^\top \bar{P}_{12} \dot{\tilde{x}}_2 + \tilde{x}_2^\top \bar{P}_{22} \dot{\tilde{x}}_2, \quad (23)$$

y mediante la sustitución de (1) en (23), se tiene:

$$\dot{V}(\tilde{x}) = k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \tilde{x}^\top \tau + k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \tilde{x}^\top \tau_h + \tilde{x}_1^\top \bar{P}_{11} \tilde{x}_2 + \tilde{x}_2^\top \bar{P}_{21} \tilde{x}_2 + \{ \tilde{x}_1^\top \bar{P}_{12} + \tilde{x}_2^\top \bar{P}_{22} \} D^{-1}(\bar{x}_1) (\tau - C(\bar{x}_1, \bar{x}_2) \tilde{x}_2 - G(\bar{x}_1)). \quad (24)$$

Aplicando ahora el principio de optimalidad de Richard Bellman, con un índice de desempeño definido como:

$$J = \frac{1}{2} \int_0^\infty \underbrace{(\tilde{x}^\top Q \tilde{x} + u^\top R u)}_{f_0(\tilde{x}, u)} dt, \quad (25)$$

y aplicando programación dinámica, es decir:

$$\min_u \left\{ \left. \frac{dV(\tilde{x})}{dt} \right|_{(10)} + f_0(\tilde{x}, u) \right\}, \quad (26)$$

de la ecuación (26), $V(\tilde{x})$ esta dada por (17), $\left. \frac{dV(\tilde{x})}{dt} \right|_{(10)}$ por (24) y $f_0(\tilde{x}, u)$ por (25), $u \triangleq \tau + \tau_h$, la matriz Q de dimensión $2n \times 2n$, verificando $Q = Q^\top$, $Q > 0$

y la matriz R de dimensión $n \times n$ es simétrica y estrictamente positiva definida. Entonces, (26) puede ser reescrita como:

$$\min_u \left\{ \begin{array}{l} k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \bar{x}_2^\top u + \bar{x}_1^\top \bar{P}_{11} \bar{x}_2 + \bar{x}_2^\top \bar{P}_{21} \bar{x}_2 \\ + (\bar{x}_1^\top \bar{P}_{12} + \bar{x}_2^\top \bar{P}_{22}) D^{-1}(\bar{x}_1) (u - C(\bar{x}_1, \bar{x}_2) \bar{x}_2 - b\bar{x}_2 - G(\bar{x}_1)) \\ + \frac{1}{2} \bar{x}^\top Q \bar{x} + \frac{1}{2} u^\top R u \end{array} \right\}, \quad (27)$$

usando las condiciones suficientes para un mínimo local, se tiene:

$$\frac{\partial}{\partial u} \left\{ \begin{array}{l} k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \bar{x}_2^\top u + \bar{x}_1^\top \bar{P}_{11} \bar{x}_2 + \bar{x}_2^\top \bar{P}_{21} \bar{x}_2 \\ + (\bar{x}_1^\top \bar{P}_{12} + \bar{x}_2^\top \bar{P}_{22}) D^{-1}(\bar{x}_1) (u - C(\bar{x}_1, \bar{x}_2) \bar{x}_2 - b\bar{x}_2 - G(\bar{x}_1)) \\ + \frac{1}{2} \bar{x}^\top Q \bar{x} + \frac{1}{2} u^\top R u \end{array} \right\} = 0, \quad (28)$$

es decir

$$k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \bar{x}_2^\top + [\bar{x}_1^\top \bar{P}_{12} + \bar{x}_2^\top \bar{P}_{22}] D^{-1}(\bar{x}_1) + u^\top R = 0 \quad (29)$$

y puesto que $x^\top y = y^\top x$:

$$k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \bar{x}_2^\top + D^{-1}(\bar{x}_1) [\bar{P}_{12}^\top \bar{x}_1 + \bar{P}_{22}^\top \bar{x}_2] + Ru = 0 \quad (30)$$

finalmente despejando el control u , se tiene:

$$u = -R^{-1} \{ k_E \tilde{\varepsilon}(\bar{x}_1, \bar{x}_2) \bar{x}_2^\top + D^{-1}(\bar{x}_1) [\bar{P}_{12}^\top \bar{x}_1 + \bar{P}_{22}^\top \bar{x}_2] \}. \quad (31)$$

4. Control del dispositivo háptico PHANToM 1.0

En esta sección se describe la arquitectura y el hardware de la interfaz háptico PNAHToM 1.0 y se obtienen las matrices de penalización Q y R del índice de desempeño definido en (25), a partir de las cuales se halla la matriz P en la ecuación algebraica de Riccati, que definen las ganancias del controlador no lineal (31). El objetivo de esta propuesta es emplear un controlador sub-óptimo no lineal diseñado para regulación en tareas de guiado háptico, por medio de la interfaz electrónica de potencia del dispositivo háptico PHANToM 1.0 y una computadora de escritorio. La interfaz electrónica de potencia del dispositivo dispone de tres Procesadores Digitales de Señales (DSP, de sus siglas en inglés), que se comunican con la computadora mediante el protocolo IEEE 1284EP en puesto paralelo, que suministra una impedancia de 199Ω a una frecuencia de 100Hz. Los actuadores de PHANToM 1.0 tienen acoplados codificadores ópticos HEDM-5500-B02, que tienen una resolución mayor a 1024 pulsos por revolución, una resolución para la medición de fuerzas de torsión de 0.03 N/mm en posición nominal, fricción de 0.004 N, fuerza máxima de 8.5 N, rigidez de 3.4 N/mm y una inercia de 75 gr y un rango de operación de -40^0 a 100^0 C, compatibles con tecnología TTL. La computadora de escritorio empleada para la evaluación del controlador (31) tiene un procesador intel core 2Duo, que trabaja a una velocidad de 2.66GHz en ambiente PC con XP y service pack 2. La programación de las estrategias de control y de la dinámica del sistema se desarrollan en Visual C++ con las herramientas H3D de Sensable Technologies, la visualización gráfica del desempeño se hace con programación fuera de línea en MATLAB 2012b.

PHANToM 1.0 es un dispositivo háptico de tres grados de libertad, con articulaciones de revolución y por ello puede ser definido como un mecanismo de eslabones articulados, que tiene propósitos distintos a los de un robot manipulador, no obstante, la técnica de modelado matemático es la misma. Su modelo dinámico se emplea para el diseño de estrategias de control y esta constituido por las ecuaciones de movimiento que relacionan las fuerzas debidas a efectos inerciales, Coriolis, centrípetas, gravitatorias y de fricción, siendo esta última despreciable por la calidad en la transmisión de movimiento articular.

De acuerdo con [13], el modelo dinámico reducido de PHANToM se obtiene mediante la formulación de Euler-Lagrange, al que se le aplica el cambio de variable $\bar{x} = x - x^+$ y definiendo $u = [u_1 \ u_2 \ u_3]^T$ y corresponden a las entradas de control para cada uno de los actuadores del sistema. Los puntos de equilibrio alrededor de los cuales se obtiene (13) y (??) se definen por $x^+ = [k_1 \ k_2 \ k_3 \ 0 \ 0 \ 0]^T$ que indican una posición articular $\theta_1 = k_1$, $\theta_2 = k_2$ y $\theta_3 = k_3$, en radianes y una velocidad 0 rad/seg. Proponiendo una linealización del sistema alrededor del punto de equilibrio $x^+ = [0.5 \ 0.5 \ 0.5 \ 0 \ 0 \ 0]^T$, con condiciones iniciales $x_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$, se tiene:

$$A = \begin{cases} a_{1,4} = 1, a_{2,5} = 1, a_{3,6} = 1, \\ a_{5,2} = 9.5839, a_{5,3} = -10.5219, \\ a_{6,2} = 37.8075, a_{6,3} = 386.5271, \\ 0 \text{ para los elementos restantes} \end{cases}, \quad B = \begin{cases} b_{4,1} = 2535, b_{5,2} = 2549, b_{6,3} = 10377, \\ 0 \text{ para los elementos restantes} \end{cases}$$

con el par (A, B) controlable. En consecuencia, se calcula un controlador alrededor del punto de equilibrio x^+ , empleando el enfoque óptimo del LQR, por medio de la función de Bellman (17) y P , que satisface la ecuación algebraica de Riccati. Para el índice de desempeño (25), se eligen las siguientes matrices Q y R

$$Q = \begin{cases} q_{1,1} = 254.1756563965411, \\ q_{2,2} = 254.1756563965167, \\ q_{3,3} = 264.4382885838342, \\ q_{4,4} = 254.9973344114833, \\ q_{5,5} = 254.9973344114585, \\ q_{6,6} = 256.0342099346721, \\ q_{ij} = 0, i \neq j, \end{cases}, \quad R = \begin{cases} r_{ij} = 6.153846153846154 \ i = j, \\ r_{ij} = 0 \ i \neq j. \end{cases}$$

que dan lugar a la siguiente P , como solución específica a la ecuación algebraica de Riccati

$$P = \begin{cases} p_{1,1} = 254.59977, p_{2,2} = 254.60165, p_{3,3} = 260.20627, \\ p_{4,4} = 0.0136556, p_{5,5} = 0.0155401, p_{6,6} = 0.0038252, \\ p_{1,4} = p_{4,1} = 0.0136328, p_{2,3} = p_{3,2} = -5.9104 \times 10^{-12}, \\ p_{2,5} = p_{5,2} = 0.0155141, p_{2,6} = p_{6,2} = 1.65698 \times 10^{-16}, \\ p_{3,5} = p_{5,3} = -3.5859 \times 10^{-16}, p_{3,6} = p_{6,3} = 0.0038874, \\ p_{5,6} = p_{6,5} = 1.383641 \times 10^{-18}, \\ 0 \text{ para los demas elementos.} \end{cases}$$

5. Resultados experimentales

En la presente sección se exhiben y discuten los resultados experimentales obtenidos de la implementación de la ley de control no lineal sub-óptima, los cuales permiten validar el rendimiento y desempeño del dispositivo PHANTOM 1.0 en una tarea de guiado háptico. El experimento realizado para reportar los resultados en esta sección corresponden a un seguimiento de trayectorias, en el cual la trayectoria deseada se define a partir de las ecuaciones paramétricas de una circunferencia sobre el plano xz , para la posición operacional y sus respectivas primera y segunda derivadas para velocidad y aceleración, como se muestra a continuación:

$$\begin{aligned} x(t) &= h + r \cos(\omega t) \\ y(t) &= 0 \\ z(t) &= k + r \sin(\omega t), \end{aligned}$$

donde $h = 0$ m., $k = 0$ m. determinan las coordenadas del centro de la circunferencia y se encuentran en el origen del plano, $r = 0.01$ m. representa el radio de la circunferencia (en metros), $\omega = 2\pi f$, la frecuencia $f = \frac{1}{ts}$ y $ts = 8$ el tiempo necesario para desarrollar la tarea un ciclo completo. La tarea de guiado se realiza durante 16 seg para observar el desempeño del dispositivo háptico al completar la trayectoria deseada en dos ciclos completos, es decir, para efectuar la trayectoria de la circunferencia dos veces, con las mismas consignas de centro y radio. Es importante mencionar que fue necesario el uso del modelo cinemático inverso de posición y del modelo cinemático inverso de velocidad, para la obtención de la posición y velocidad articular respectivamente, que describen las posiciones y velocidades operacionales deseadas, dado que el control que presentado en la sección 3 esta diseñado en el marco articular.

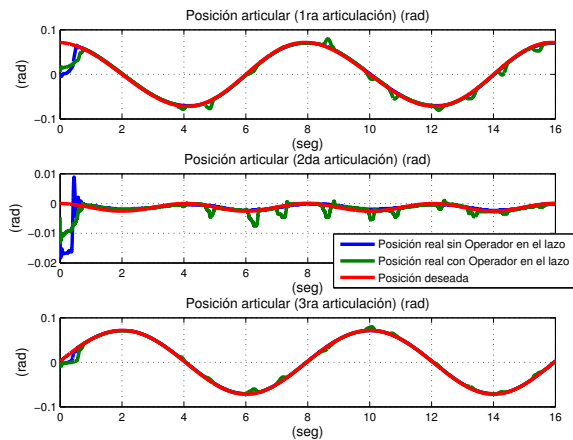


Figura 1. Posición articular y referencias deseadas.

La Figura (1) representa la posición articular con y sin operador en el lazo de control, en la cual se observa un seguimiento articular con alto desempeño para ambos casos, donde la posición articular real converge a la deseada en un tiempo menor a 0.5 seg. y permanece con este comportamiento a lo largo del tiempo de ejecución, y en el experimento con operador en el lazo se muestra como el control propuesto lleva al operador a desarrollar la trayectoria deseando aun con las perturbaciones inducidas por el operador.

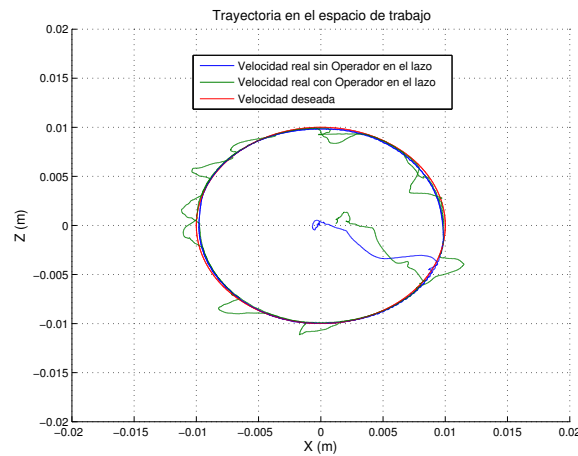


Figura 2. Trayectoria en el espacio de trabajo con propósito de entrenamiento Kinestético.

La Figura (2) ilustra la trayectoria del dispositivo háptico PHANTOM 1.0 sobre su espacio de trabajo, bajo la acción del control (31), en ella se aprecia la correcta conversión de variables articulares a variables operacionales por medio de los modelos cinemáticos, también es apreciable que el error en el marco operacional esta por de bajo del orden de los milímetros, pues la circunferencia deseada tiene radio equivalente a un centímetro, en esta misma Figura, se observan las perturbaciones inducidas por el operador en el guiado háptico y como el control se adapta a ellas para continuar con el seguimiento dadas las dinámicas no moledadas introducidas por el comportamiento del operador con baja experiencia en guiado, razón que justifica el alto desempeño del controlador propuesto para tareas de guiado háptico que requieren de alta precisión.

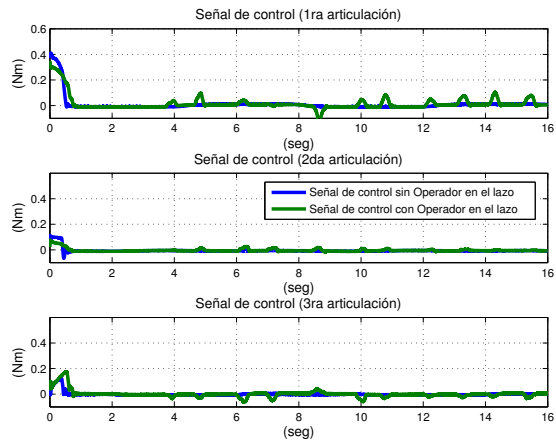


Figura 3. Señal de control aplicada.

La contribución de la estrategia de control propuesta adquiere mayor evidencia en la Figura (3), puesto que el dispositivo logra convergencia en intenta la trayectoria deseada con la energía empleada, en movimiento libre y en presencia del operador con baja experiencia en este tipo de tareas respectivamente, la energía para ambos casos es relativamente baja y con valores máximos en el orden de los 0.5 Nm., como se aprecia en el caso de la primera articulación.

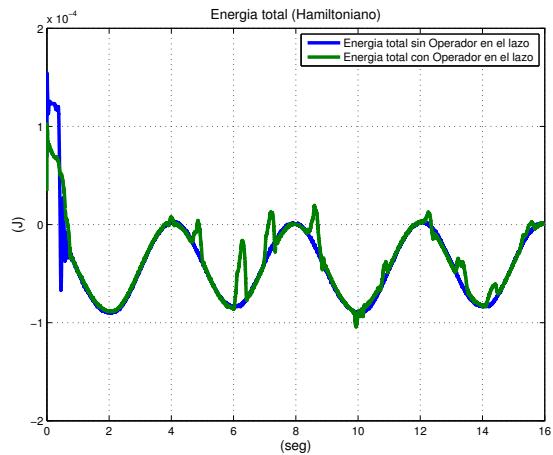


Figura 4. Energía total empleada en la tarea, con el operador en el lazo en condiciones de disturbio.

Las tendencias de la magnitud en la señal de control de la Figura (3) en los primeros instantes del experimento para el caso de movimiento libre se encuentran directamente relacionados con las condiciones iniciales del dispositivo y las condiciones deseadas para la tarea definida, lo cual abre la posibilidad de reducir la magnitud del error tanto en posición y velocidad, como magnitud de la señal de control, por medio de una colocación del efecto final directamente sobre un punto de la trayectoria deseada, antes de comenzar el guiado. Esta propuesta no da a conocer estudios comparativos con otras técnicas de control, sin embargo es notable que la energía empleada por dispositivo con otras estrategias de control como es el caso de [7,11], es comparativamente mayor a la empleada en nuestra propuesta

Por ultimo, la Fig. (4) ilustra el comportamiento de la energía total empleada en la tarea de guiado háptico bajo la acción del controlador no lineal sub-óptimo en movimiento libre y bajo las perturbaciones inducidas por el operador humano en el lazo de control, en ella puede apreciarse que la cantidad de energía requerida para la tarea de guiado es relativamente mínima, en particular es definida solo por energía potencial del sistema para el primer caso, y con la adición de la energía aplicada por el operador dadas las restricciones como el conocimiento incierto de la trayectoria deseada y la baja experiencia en tareas guiado háptico.

6. Conclusiones

El controlador sub-óptimo que se ha presentado en este artículo, para el seguimiento de trayectorias y el desarrollo de tareas en guiado háptico pasivo, ha tenido resultados experimentales con desempeño satisfactorio y excelente manipulabilidad dinámica con consumo de energía relativamente bajo. La principal contribución de esta propuesta y de sus resultados, ilustran la posibilidad de métodos de control no lineal y sub-óptimos para la solución de problemas de minimización del consumo de energía en el control de robots manipuladores, para minimizar el esfuerzo del mecanismo de eslabones articulados e interfaces electromecánicas de potencia, preservando la vida útil de sus mecanismos, y la obtención de un comportamiento apropiado para tareas de guiado háptico mediante la penalización de la convergencia del estado y la magnitud de la señal de control en el índice de desempeño. Se demuestra como una función cuadrática como (19), es útil en la síntesis de controladores no lineales sub-óptimos. Se aprovecharon las propiedades del modelo del dispositivo háptico para sintetizar la estrategia de control no lineal propuesta. El algoritmo de control no lineal sub-óptimo se considera factible para su implementación en aplicaciones de robots manipuladores e interfaces hápticas que requieren de alto desempeño y bajo consumo monitoreado de la energía. Los experimentos presentados demuestran un rendimiento aceptable del dispositivo bajo la ley de control penalizada adecuadamente, propuesta para tareas de guiado háptico.

Agradecimientos. El primer autor agradece a **CONACYT** por la beca de maestría otorgada durante el periodo Agosto 2011- Junio 2013, con número de registro **290649**.

Referencias

1. Athans M., Falb P.L. *Optimal Control: An Introduction to the Theory and Its Applications*, Mc Graw Hill, New York, 1966.
2. Basin M., Sanchez E., Martínez-Zúñiga R. Optimal filtering for systems with multiple state and observation delays. *Int. Journal of Innovative Computing, Information and Control*, vol. 5, pp.1309-1320, 2007.
3. Burdea, G.C. *Force and Touch Feedback for Virtual Reality*, Ed. John Wiley and Sons, Inc., Electrical and Computer Engineering Department, The State University of New Jersey, 1996.
4. Hemami, A. Mehrabi, M. G. and Cheng, R. M. H. Synthesis of an optimal control law for path tracking in mobile robots, *Automatica*, vol. 28(2), pp. 383-387, 1992.
5. Goldstein, H. *Classical Mechanics*. Ed. Englewood Cliffs, NJ: Prentice-Hall, 1990.
6. Kalman, R.E. Contributions to the theory of optimal control. *Bol Soc Mat Mex*, vol. 5, pp 102-109, 1960.
7. Domínguez-Ramírez, O. A. & Parra-Vega, V. Active Haptic Interface with Purposes of Remote Training, *Proceedings of the 11th International Conference on Advanced Robotics*, ISBN 972-96889-9-0, vol. 3, pp. 1588-1593, University of Coimbra, Coimbra, Portugal, 2003.
8. Hrovat D. Survey of advanced suspension developments and related optimal control applications, *Automatica*. vol 33. no. 1, pp. 1782-1817, 1997.
9. Lin, F. and Brandt, R.D. An optimal control approach to robust control of robot manipulators, *IEEE Transaction on Robotics and Automation*, vol. 14(1), pp 69-77, 1998.
10. Padhi. R., Prabhat P., Balakrishanan S.N., Reduced order optimal control synthesis of a class of nonlinear distributed parameter systems, using single network adaptive critic design, *Int. Journal of Innovative Computing, Information and Control*, vol.4, no.2, pp.457-470, 2008.
11. Parra-Vega, V., S. Arimoto, Y. H. Liu, G. Hirzinger, and P. Akella, Dynamic Sliding PID Control for Tracking of Robot Manipulators: Theory and Experiments, *IEEE Transaction on Robotics and Automation*, Vol. 19 (6), pp 967-976, December, 2003.
12. Ordaz-Oliver J.P., Santos-Sánchez O.J. López-Morales V., On the Sub-optimal Feedback Control Law Synthesis of Underactuated Systems *International Journal of Innovative Computing, Information and Control*, vol.5, no.9., pp2791-2808, 2009.
13. Domínguez-Ramírez, O. A. and Parra-Vega, V. Realistic Active Haptic Guided Exploration with Cartesian Control for Force-Position Tracking in Finite Time. *Journal of Applied Bionics and Biomechanics*, ISSN 1176-2322, ABBI, Vol 3 (4) pp 279-289, Cambridge, England, 2006.
14. Verscheure, D., Demeulenaere, B., Sweveres, J. De Schutter, J. and Diehl, M. Time optimal path tracking for robots: a convex optimization approach, *IEEE Transaction on Automatic Control*, vol. 54(10), pp. 2318-2327, 2009.

Índice de autores

Águila Rodríguez, Gerardo	447	González-Marrón, David	343
Aguilar-Lasserre, Alberto A.	437	González-Mendoza, Miguel	181, 281, 335, 343, 361
Alejo, Roberto	15	González-Sánchez, Blanca E.	437, 447
Alemán, Yuridiana	139	Guzmán Arenas, Adolfo	335
Alvarado-Uribe, Joanna	343	Hernández Gress, Neil	281
Arzate Cruz, Christian Jesús	189	Hernández Palacios, Raúl	427
Baltazar, Rosario	233, 243	Ibarra-Orozco, Rodolfo	181
Bandyopadhyay, Sivaji	79	Jiménez-Salazar, Héctor	69, 109
Barragán-López, Darío	3	Lechuga G., Luis R.	295
Briseño Cerón, Abraham	417	Leon Martagón, Gilberto	69
Buzón Cantera, Ivo Erasmo	373	López-Pimentel, Juan Carlos	181
Calzada, Valentín	197	Loya, Nahun	119
Camarillo-Ramírez, Pablo	129	Mancilla Espinoza, Luis	243, 405
Camiña Prado, J. Benito	45, 57	Marcial-Romero, J. Raymundo	15
Carpio, Martín	197, 243, 405	Marín Hernández, Antonio	161
Carvajal-Gámez, Blanca Esther	209	Martínez-Sibaja, Albino	437
Castro-Espinoza, Félix	25, 305	Maximiliano Méndez, Gerardo	259
Conde-Ramírez, J. Carlos	129	Meléndez, Acosta, Nayeli Joaquinita	161
Coria, Sergio R.	3	Mondragón-Becerra, Rosibelda	3
Cornelio Martínez, Patricia Ivette	437, 447	Monroy, Raúl	45, 57
Cortes-García, Patricia	25	Montaño-Murillo, Roberto A.	437, 447
Cuesta-Frau, David	233	Montes Dorantes, Pascual Noradino	259
de Anda, Juan	243	Mora Vargas, Jaime	373
De Gyves, Oriam	319	Nieto González, Juan Pablo	259
Díaz-Torres, Víctor O.	109	Núñez Cárdenas, Felipe de Jesús	427
Domínguez-Ramírez, Omar A.	295, 393, 417, 469	Ochoa Gracia, Eduardo Emmanuel	459
Enciso-González, Angélica	343	Olmos, Iván	119
Fernández R., José M.	295	Ordaz-Oliver, Mario O.	393, 469
Franco-Arcega, Anilu	25	Oregel, Josué	405
Gallegos-Funes, Francisco Javier	209	Ornelas, Manuel	197, 243, 405
García Calderón, Daniel	361	Padilla-Navarro, Christian	233
García, Juan	171	Pakray, Partha	79
Gelbukh, Alexander	79	Parra Vega, Vicente	417
González Sánchez, Blanca Estela			
González, Jesús	119		
González, Aram B.	271		

Pedraza Martín, Armando	305	Sánchez García, Angel Juan	151
Pedroza Méndez, Blanca E.	459	Sánchez Pulido, Miquelina	459
Pérez Ramírez, Rosa Elena	35	Sánchez-López, Abraham	129
Pérez-Meza, Mónica	3	Sánchez-Sánchez, Christian	69, 109
Pinto, David	119, 139	Santiago, Raúl	197
Posada-Gómez, Rubén	437, 447	Santibáñez, Mónica	15
Pozas Cárdenas, Mariano	427	Santos-Sánchez,	
Praga Alejo, Rolando	259	Omar J.	253, 305, 393, 469
Puga, Héctor J.	197, 243, 405	Sidorov, Grigori	95
Ramírez Cruz, José Federico	459	Soriano, Patrick	373
Ramírez Uresti, Jorge A.	189, 271	Sotelo, Marco	243
Ramírez Z., Juan D.	295	Tabarez Paz, Israel	281
Ramos V., Luis E.	295	Toledo, Leonel	319
Ramos-Fon Bon, Víctor	181	Tomás Mariano, Víctor Tomás	427
Rendón, Erendira	15	Trejo-Salazar, David	209
Rios Figueroa,		Uribe H., Miguel O.	295
Homeró Vladimir	151, 161	Valdovinos, Rosa María	15
Rivalcoba Rivas, Jorge Ivan	219	Velasco Bermeo, Nestor	361
Rivera González, Martha Idalid	335	Venegas, Salvador	171
Rodríguez Vega, Dora A.	417	Vilariño, Darnes	139
Rodríguez-Ruiz, Jorge	45	Villatoro-Tello, Esaú	69, 109
Rojas-Rauda, Rosa-Irene	253	Zamora Ginez, Irma del Carmen	
Rudomín, Isaac	189, 219, 319	Zamudio-Rodríguez, Victor	233, 405
Ruiz Bartolomé, Ángel	373	Zepeda Cortés, Claudia	35

Comité editorial del volumen

Moises Alencastre-Miranda
Ildar Batyrshin
Felix Castro Espinoza
Noé Castro Sánchez
Jair Cervantes
Anilu Franco Arcega
Juan Frausto-Solis
Sofia N. Galicia-Haro
Ariel Lucien Garcia Gamboa
Alexander Gelbukh
David Gonzalez
Miguel Gonzalez-Mendoza
Neil Hernandez Gress
Oscar Herrera
Rodolfo Ibarra-Orozco
Antonio Marin Hernandez
Luis Felipe Marin Urias
Raul Monroy
Jaime Mora-Vargas
Lourdes Muñoz Gómez
Antonio Neme
Elvia Palacios
Obdulia Pichardo Lagunas
Jorge Adolfo Ramirez Uresti
Dafne Rosso
Paolo Rosso
Grigori Sidorov
Israel Tabarez Paz
Raul A. Trejo-Ramírez
Benjamin Valdes
Edgar Vallejo
Nestor Velasco
Alfredo Villagran

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
mayo de 2013
Printing 500 / Edición 500 ejemplares