RESEARCH IN COMPUTING SCIENCE

ISSN: 1870-4069

Special issue:
Advances in Pattern Recognition

Jesús Ariel Carrasco Ochoa
José Francisco Martínez Trinidad
Juan Humberto Sossa Azuela (Eds.)

Vol. 44

Special Issue:
Advances in Pattern Recognition

Research in Computing Science

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 44**, Noviembre, 2009. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, 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: *Juan Humberto Sossa Azuela*, RFC *SOAJ560723*

***Research in Computing Science** is published by the Center for Computing Research of IPN. Volume 44, November, 2009. Printing 500. Authors are responsible for the contents of their papers. 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, November, 2009, in the IPN Graphic Workshop – Publication Office.*

Volume 44
Volumen 44

Special Issue:
Advances in Pattern Recognition

Volume Editors:
Editores del Volumen

Jesús Ariel Carrasco Ochoa
José Francisco Martínez Trinidad
Juan Humberto Sossa Azuela

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2009

ISSN: 1870-4069

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

Printing: 500
Impresiones: 500

Printed in Mexico
Impreso en México

Preface

This special issue presents original research papers on Pattern Recognition, Image and Document Analysis, and real word applications. These papers were accepted to be presented at the first Mexican Workshop on Pattern Recognition MWPR 2009. MWPR2009 was created as a forum for exchanging scientific results and experiences, as well as sharing new knowledge, and increasing the co-operation between research groups in pattern recognition and related areas, in Mexico. This first workshop was organized by the Computer Science Department of the National Institute of Astrophysics, Optics and Electronics (INAOE) and the Mexican Association for Computer Vision, Neural Computing and Robotics (MACVNR). MWRP2009 was sponsored by the Research and Advanced Studies Center campus Guadalajara (CINVESTAV-GDL), and the Center for Computing Research of the National Polytechnic Institute (CIC-IPN).

A total of 26 full papers were received for evaluation. Each submission was reviewed by at least two independent members of the Editorial Board of the volume. This volume contains revised versions of 18 papers, selected for publication after thorough evaluation. Thus the acceptance rate was of 69%.

We cordially thank all people involved in the preparation of this volume. In the first place, to the authors of the papers, whose contributions allowed the success of the MWPR2009. We also thank the members of the Editorial Board of the volume and additional referees. The submission, reviewing and selection process was supported for free by the EasyChair system, www.EasyChair.org.

We hope this first edition of the Mexican Workshop on Pattern Recognition become the starting point for a series of successful events involving the Mexican Pattern Recognition Community.

Jesús Ariel Carrasco Ochoa
José Francisco Martínez Trinidad
Juan Humberto Sossa Azuela

November 2009

Table of Contents

Índice

Page/Pág.

Computer Vision

Language and Document Processing

Image Processing and Texture Analysis

Pattern Recognition Techniques

Computer Vision

Robust Extrinsic Camera Calibration from Trajectories in Human-Populated Environments

Guillermo Baqueiro and Jean-Bernard Hayet

Computer Science Group,
Centro de Investigación en Matemáticas (CIMAT, A.C.),
Guanajuato, Gto. 36240 México.
baqueiro,jbhayet@cimat.mx *

Abstract. This paper proposes a novel robust approach to perform inter-camera and ground-camera calibration in the context of visual monitoring of human-populated areas. By supposing that the monitored agents evolve on a single plane and that the cameras intrinsic parameters are known, we use the image trajectories of moving objects as tracked by standard trackers in a RANSAC paradigm to estimate the extrinsic parameters of the different cameras. We illustrate the performance of our algorithm on several challenging experimental setups.

1 Introduction

In spite of its spectacular development, video-surveillance is yet largely depending on human agents in charge of monitoring up to dozens of TV screens, which may be responsible for negative detections. Recent years have seen the emergence of automatic, computer-aided video-surveillance systems in the computer vision community. Typically these systems use state-of-the-art tracking algorithms in each camera in the network, and fusion techniques to recover 3D trajectories of the moving objects in the scene [1]. Then, this information feeds pre-defined or unusual event detection and may trigger alarms. An important element for a widespread use of such systems is an *automatic* calibration algorithm that would not require the costly intervention of an expert.

This article presents such an algorithm, that estimates the extrinsic parameters of different cameras involved in a surveillance network, i.e. the different 3D transformations between pairs of cameras and between each camera and the reference plane. The assumptions we make are that (1) the targets are moving on a planar scene, which is a common setup in surveillance systems, (2) we have an estimation of the intrinsic parameters of the cameras, and (3) the cameras are static. An important characteristic of such camera networks is that the viewpoints may be dramatically different from one camera to another, e.g., in the frames from two sequences of cameras at Fig. 1. In particular, this prevents us from using traditional feature-based matching techniques based on local descriptors around interest points [2] for estimating the underlying geometric transforms. Instead, in the vein of the seminal work of [3], we rely on the output of motion detection and motion tracking to guess correspondences at the level of motion blobs or motion tracks and to infer the corresponding geometry.

* This work was partially funded by CONCyTEG through its grant 09-02-K662-073.

© J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad and H. Sossa (Eds.)
Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 3-10

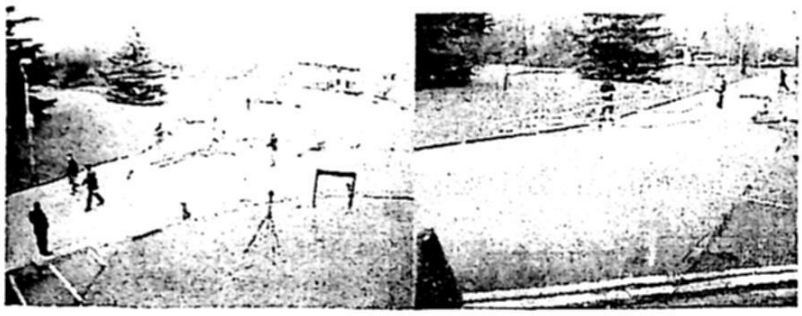


Fig. 1. The typical input/output of our algorithm: we form correspondences (one color, one correspondence) among trajectories (from standard trackers) to compute robustly the feet-to-feet homography H_{ij} between the two views. These inlier *trajlets* (see Section 3) correspond to the computations of homographies between camera 2 and 7 of the PETS 2009 data (i.e., subfigures of the right column of Fig.3).

The organization of this paper is as follows: In Section 2, we highlight noticeable related work in the literature; in Section 3, we describe our algorithm for robust inter-camera homography estimation; in Section 4, we recover all the extrinsic parameters from homographies and in Section 5 we comment results on different setups; finally, Section 6 draws conclusions and introduces future work.

2 Related work and contributions

The seminal work of Lee and al. [3] uses the centroids of blobs extracted with standard background subtraction techniques to perform homography fitting with a least median square (LMS) approach, that is further refined in a second step. Its main drawback is that the number of putative correspondences grows very fast with the number of targets, so that the number of inliers for the LMS optimization drops dramatically in proportion, making the algorithm unsuitable for regularly crowded scenes. Obviously the dimension of the search space is reduced drastically when, instead of *motion detection blobs*, one form the correspondences from *tracking sequences* [4, 5]. In [5], the authors present a RANSAC-like approach that performs non-uniform sampling in the set of putative sequences. It sequentially tests homographies from two pairs of sequences (two pairs in each video) and keeps the best homography but the likelihood functions that ponder each sample are not clearly defined. The work in [4] is more general in a sense, as it is extended to fundamental matrix estimation, and is also based on RANSAC, but does not make particular distinction between samples to guide the consensus to the most promising pairs of sequences. In another paradigm, the work of [6] uses perspective invariants, namely the cross ratio of five points, to select corresponding trajectories between video sequences. The algorithm also allows to calibrate the time offset between cameras. However, in most situations, it is quite difficult to isolate non-degenerate trajectories – i.e, sufficiently far from straight lines – to compute stable cross-ratios, so that the possible applications of this work are limited. A common inconvenient of the previous approaches is that it uses tracking trajectories as they come from the tracking algorithm, which causes problems of robustness in the case that the tracking fails – and that the system is not aware of it. Among the most recent works in the area, the one

of [7] is interesting as it also takes radial distortion into account. However, the correspondences are determined on the base of control points manually selected onto trajectories, which may make it more adapted for expert users.

In many situations, e.g. because of occlusions in crowded scenes, tracking algorithms may be confounded and may assign a wrong identity to some tracked object. This may be catastrophic for the estimation of scene geometry. Our approach brings several contributions among which (1) robustness with respect to the possible failures of the tracking algorithms and (2) more reliable guidance of the optimization process to the correct geometry.

3 Robust inter-video homography estimation

3.1 Problem formulation

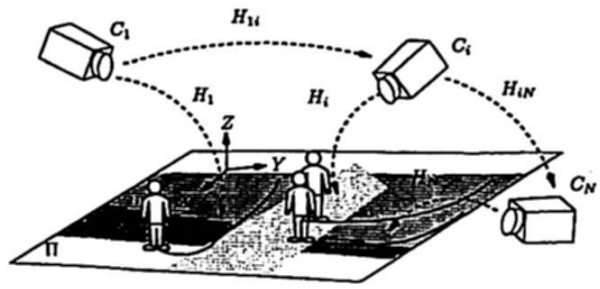


Fig. 2. Setup: Several cameras C_i for $1 \leq i \leq N$ observe a *planar* scene laid on a plane Π. The inter-camera homographies between two cameras i and j are denoted by H_{ij}, while the camera-to-reference plane homographies from camera i to Π are denoted by H_i. Each camera has it own (shaded) scene coverage.

The problem setup and notations are detailed in Fig. 2: several cameras C_i, $1 \leq i \leq N$, with different degrees of overlap, monitor a scene where people or other mobile objects move. We suppose that this scene is laid on a reference plane Π, which induces an *homography* between any pair of cameras (i, j) monitoring the scene, i.e. if $p_i = (u_i, v_i)^T$ is an image point in camera C_i, projection of a point P of the plane Π, and $p_j = (u_j, v_j)^T$ the projection of this same point on camera C_j, then we have the classical relationship [8],

$$p_i = \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \sim H_{ij} p_j = \begin{pmatrix} h_{ij}^{11} & h_{ij}^{12} & h_{ij}^{13} \\ h_{ij}^{21} & h_{ij}^{22} & h_{ij}^{23} \\ h_{ij}^{31} & h_{ij}^{32} & h_{ij}^{33} \end{pmatrix} \begin{pmatrix} u_j \\ v_j \\ 1 \end{pmatrix}, \tag{1}$$

where $\sim$ means that a relation of equality holds for any multiplication factor $\lambda > 0$, so that H_{ij} has in fact only 8 degrees of freedom. The problem consists in estimating these transforms, first, and, on a second step, the homographies H_i that maps points p_i to the points they are the projection of, i.e. points P,

$$p_i = \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \sim H_i P = \begin{pmatrix} h_i^{11} & h_i^{12} & h_i^{13} \\ h_i^{21} & h_i^{22} & h_i^{23} \\ h_i^{31} & h_i^{32} & h_i^{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}, \tag{2}$$

where $P = (X, Y, 0)^T$ are the coordinates of point P in a frame (X, Y, Z) (depicted in Fig. 2) such that $Z = 0$ is the equation of Π. Traditional methods estimate homographies H_{ij} by searching for point correspondences (p_i, p_j) and using them to solve the linear system directly induced by Eq. 1. As these correspondences are extremely difficult to find with points and appearance whenever the viewpoint changes strongly, we rely on tracks from video trackers.

Our algorithm can be summarized in: (1) **collect trajectories** in each stream V_i with a tracking algorithm, (2) **pre-process** the trajectories to eliminate potential ambiguities at occlusion points, we will refer to the trajectory parts built in this way as *trajlets*, and (3) apply the RANSAC-like robust **optimization process** with a likekihood-guided sampling process.

3.2 Collecting and pre-processing trajectories

As we want the algorithm to be *robust* w.r.t. the properties of the 2D tracker, we apply any state-of-the art 2D tracker on our video streams and collect the resulting tracks. Practically, we used different algorithms implemented in the OpenCV library. The result, for a video stream i of camera C_i, is a set of trajectories $L_i = \{l_i^{(m)}, m \geq 0\}$, encoding the position of one target **centroid** along the time. The centroids are chosen because they are not as sensitive to noise as the feet position. However, a consequence is that the computed homography will correspond to a *plane passing through target centroids* (i.e., not Π). We denote it as $\hat{H}_{ij}$ and will see that H_{ij} can be deduced from $\hat{H}_{ij}$.

The second step forms what we call *trajlets*, i.e., pieces of trajectories that are a priori not susceptible to be contaminated by occlusion errors from the tracking algorithm. For all pairs of trajectories $(l_i^{(m)}, l_j^{(m)})$ for which some of the points $p_{i,t}, p_{j,t}$ are close (at some timestamp t), we simply cut off the ambiguous parts on a given time radius δ. This creates four sub-trajectories (*trajlets*) $(l_i^{(m)+}, l_t^{(m)-}, l_j^{(n)+}, l_j^{(n)-})$, such that,

$$\begin{cases} l_i^{(m)} = l_i^{(m)-} \cup \{p_{i,t-\delta} .. p_{i,t+\delta}\} \cup l_i^{(m)+}, \\ l_j^{(n)} = l_j^{(n)-} \cup \{p_{j,t-\delta} .. p_{j,t+\delta}\} \cup l_j^{(n)+}. \end{cases}$$

Also, we smooth these trajectories by using local filtering based on Bezier curves. The result of this processing is, again, for each video stream i, a – a priori larger – set $L_i' = \{l_i^{(m)}, m \geq 0\}$, as illustrated by Fig. 1.

3.3 Robust homography estimation

The estimation of $\hat{H}_{ij}$ is done in a RANSAC-like scheme described in this section. A priori, a candidate homography for explaining the two images from the same scene can be derived from just one correspondence between a trajectory in i and a trajectory in j, since it is entirely defined by 4 point correspondences [4, 6]. However, most of the *trajlets* appearing in usual video-surveillance contexts are close to degenerate, i.e. linear. This is why we generate here the candidate homographies from *two trajlets* correspondences instead of one. This, in turn, has an inconvenient, since, if we have an order of magnitude of τ *trajlets* appearing at intersecting windows of time, then the probability for a sampled pair (i, j) to

match is $\frac{1}{\tau}$, and the one for two consecutively sampled trajlets $\frac{1}{\tau^2}$. Hence, the number of samples needed in RANSAC to ensure (in expectation) that a correct pair is sampled is **quadratic** in τ, which can be problematic with crowded scenes.

A solution is to avoid an uniform sampling process by assigning likelihood values to all possible pair of trajectories, and by sampling the trajectories pairs according to these values of likelihoods. We define them in the following way:

$$p(l_i^{(m)}, l_j^{(n)}) \propto \frac{1}{\max(N_j(l_i^{(m)}), N_i(l_j^{(n)}))} \Delta(l_i^{(m)}, l_j^{(n)}),$$

where $N_j(l_i^{(m)})$ stands for the number of *trajlets* from L'_j that have a time overlap with trajectory $l_i^{(m)}$ (the value being defined as one if there is no time overlap) and $\Delta(l_i^{(m)}, l_j^{(n)})$ measure the time overlap. These two terms (1) penalize the sampling of trajectories that could result **ambiguous to match** (large N_j or N_i) and (2) favor those trajectories with large overlap, which should improve the homography computation by preferring larger sequences. Both criteria make the required number of samples much lower than the aforementioned quadratic term above. In practice, we need a few dozens iterations to get a pair of correctly matched *trajlets*. The whole optimization process is described in Algorithm 1.

Algorithm 1 Homography computation between cameras C_i and C_j

$\hat{S} \leftarrow 0$
repeat
1. Sample a pair of *trajlets* $(l_i^{(m)}, l_j^{(n)})$ according to the likelihoods $p(l_i^{(m)}, l_j^{(n)})$.
2. Compute the candidate homography $\hat{H}_{ij}^{mn}$ from the correspondences between all points of $l_i^{(m)}$ and $l_j^{(n)}$), by using the classical DLT [8], and its inverse $(\hat{H}_{ij}^{mn})^{-1}$.
3. For all trajlets pairs $(l_i^{(r)}, l_j^{(s)})$ compute the residual symmetric error $\epsilon^2(r,s)$:

$$\epsilon^2(r,s) = \frac{1}{2|l_i^{(r)} \times l_j^{(p)}|} \sum_{(p_r,p_s)\in l_i^{(r)}\times l_j^{(p)}} d^2(\hat{H}_{ij}^{mn} p_r, p_s) + d^2(p_r, (\hat{H}_{ij}^{mn})^{-1} p_s).$$

4. Identify in the residual matrix $\epsilon^2(r,s)$ elements that are (1) below a given threshold and (2) minima on the line r and column s.
5. Sum in S the lengths of the trajectories corresponding to the identified elements.
6. If $S > \hat{S}$, $\hat{S} \leftarrow S$; $\hat{H}_{ij} \leftarrow \hat{H}_{ij}^{mn}$.

until a given proportion of trajectories from video streams V_i and V_j have been explained by $\hat{H}_{ij}$ or a given number of iterations have been done.
if $\hat{H}_{ij}$ explains a sufficient proportion of trajectories in V_i and V_j **then**
  consider $\hat{H}_{ij}$ as recovered
else
  consider the two views as un-registered.
end if

4 Extrinsic parameters estimation

Homography decomposition. Once the homographies $\hat{H}_{ij}$ have been recovered, we estimate extrinsic parameters, i.e. the parameters of the rigid 3D transform between the two cameras acquiring video streams i and j. To this purpose, we use the following decomposition of H_{ij} [8],

$$\hat{H}_{ij} \sim K_i[dR + tn^T]K_j^{-1} \tag{3}$$

where the matrices K_i are the intrinsic parameters of cameras i, supposed known here, and where d, n give the equation of the plane (here, the centroids plane) in camera i frame, i.e. its equation is $n^T Q = d$, where Q are the coordinates of 3D points in the camera i frame. Note that Eq. 3 is given only *up to a scale factor*, that we will determine in a second time. We use Triggs' algorithm to determine the decomposition values [9]. It gives two possible pairs (R, t), one of them being easily discardable.

Image plane to ground plane homography. From the image-to-image homography and its decomposition, one recovers[1] an homography to the centroids plane, by deriving from the projection equation on camera C_i,

$$(u_i, v_i, 1)^T \sim K_i(dn + Q_{n^\perp}),$$

where $Q_{n^\perp}$ is the component of Q onto the centroids plane. By choosing a base e_1, e_2 of vectors generating the centroids plane, for example $e_1 = (1, 0, 0)^T \wedge n$, and $e_2 = e_1 \wedge n$, one derives in terms of the spatial coordinates on the – real – centroids plane, (α, β),

$$(u_i, v_i, 1)^T \sim K_i \left(e_1, e_2, dn\right) (\alpha, \beta, 1)^T,$$

i.e. $\hat{H}_i = K_i \left(e_1, e_2, dn\right)$ acts as an homography from the centroids plane to the image plane in camera C_i.

Scale recovery. As t and d are computed only up to a scale, we used some knowledge about the scene to compute the scale factor, one option is to assume a constant, fixed velocity for the object in the scene with median velocity, another one to assume a half-height of one meter between people's centroids and feet. In our experiments, the second option gave better and much stabler results[2]. Once the scale is recovered, one finally gets either a ground plane-to-image H_i (as in Eq. 2) or an image-to-image H_{ij} induced by the ground plane Π (as in Eq. 1). Most of the results below illustrate this second form. Note that for the moment all the homographies are expressed in a different frame relative to Π.

5 Results

As mentioned before, this work has been implemented in C++ with the OpenCV library. We tested our algorithm on the PETS [10] benchmark data, which provides different case studies, with eight video streams for each case, corresponding

[1] Up to a scale factor for d !

[2] It seems that one problem in using velocity information is that the framerates of most videos of the benchmark data are not constant.

to different level of people density. For calibration purposes, we used the medium density crowd dataset (S0), and give some results in Fig. 3. We depict three of the computed homographies H_{ij} (left) and their inverse (right) by warping the image j onto the image plane i. It can be observed that most of the spottable elements in Π (roads, lines) that appear in both views are mostly correctly warped in the other view. The Fig. 1 gives the inliers *trajlets* that served to the estimation of the homography of the last column (i.e. between views 2 and 7). In addition of the three examples shown in Fig. 3, the algorithm can calibrate 15 of the possible 28 camera pairs, and we are still improving the algorithm to get more homographies (in fact some of the cameras have too little overlap).

6 Conclusion

The external calibration algorithm we present in this paper exhibit several advantages over comparable algorithms in the literature: (1) by using pieces of trajectories adequately sub-divided and by assigning them likelihood values reflecting a priori values for being matched unambiguously, it keeps the computational complexity (driven by the number of iterations to perform in the RANSAC-like process) reasonable; (2) the fact of not relying on the entire trajectories, but instead in smaller parts, makes it much more robust to occlusions that occur with 2D tracking algorithms. We presented rectification results in some challenging situations, where the viewpoint changes make nearly impossible to recover the geometrical transform by traditional point correspondences techniques.

Our ongoing and future work focus in determining uncertainties on the different homographies we compute, and by using these uncertainties to (1) reoptimize all homographies by using relationships of the form $H_{ij} = H_{ik}H_{kj}$ and (2) track moving in objects in 3D, both in a probabilistic manner.

References

1. Du, W., Hayet, J.B., Verly, J., Piater, J.: Ground-target tracking in multiple cameras using collaborative particle filters and principal axis-based integration. IPSJ Transactions on Computer Vision and Applications **1** (2009) 58–71
2. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2003) 257–263
3. Lee, L., Romano, R., Stein, G.: Monitoring activities from multiple video streams: Establishing a common frame. IEEE Trans. on Pattern Analysis and Machine Intelligence **22**(8) (August 2000) 1–10
4. Caspi, Y., Simakov, D., Irani, M.: Feature-based sequence-to-sequence matching. Int. J. Comput. Vision **68**(1) (2006) 53–64
5. Stauffer, C., Tieu, K.: Automated multi-camera planar tracking correspondence modeling. IEEE Conf. in Computer Vision and Pattern Recognition (CVPR'03) **1** (2003) 259
6. Nunziati, W., Sclaroff, S., Del Bimbo, A.: Matching trajectories between video sequences by exploiting a sparse projective invariant representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **PP**(99) (2003) 1
7. Kayumbi, G., Cavallaro, A.: Multi-view trajectory mapping using homography with lens distortion correction. EURASIP J. on Image and Video Processing (2008)
8. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
9. Triggs, B.: Autocalibration from planar scenes. In: In Proc. ECCV. (1998) 89–105

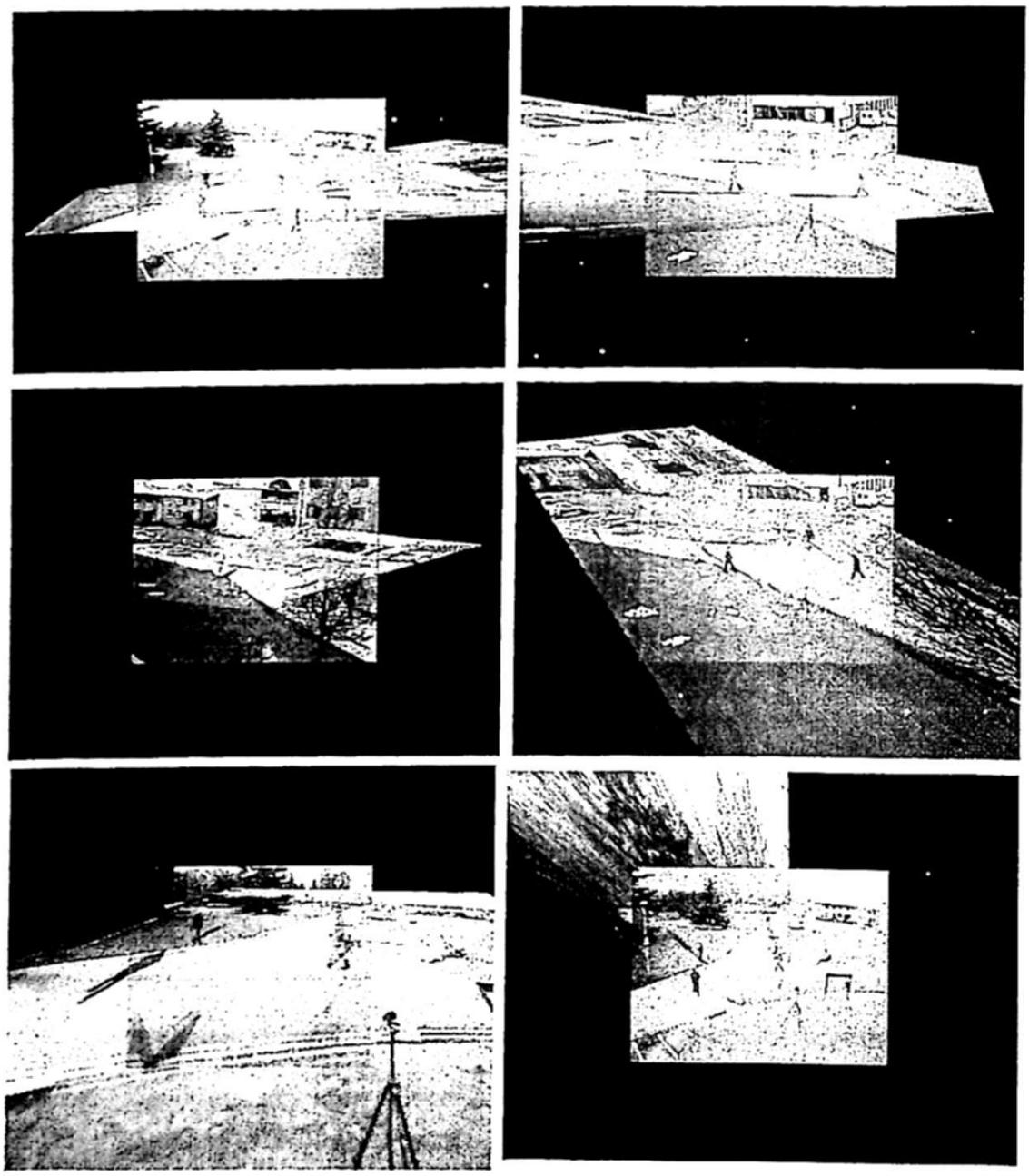


Fig. 3. Video registrations on the PETS2009 [10] sequences. Each column depicts one pair (i, j) and the corresponding homography H_{ij} (up) and its inverse (down). The processed pairs are, from left to right, $(1, 2)$, $(1, 3)$ and $(2, 7)$.

10. : Ieee int. workshop on performance evaluation of tracking and surveillance. http://www.cvg.rdg.ac.uk/PETS2009/ (2009)

Corner Detection Using Kernel Windows Matching For Scene Recognition

Mario I. Chacon-Murguia[1], Cynthia P. Guerrero-Saucedo[2], Rafael Sandoval-Rodriguez[2]

[1] DSP & Vision Laboratory, [2]Robotic Laboratory
Chihuahua Institute of Technology, Chihuahua, Mexico
{mchacon, cpguerrero, rsandova}@itchihuahua.edu.mx

Abstract. Corner detection is an important step in applications of object recognition, scene recognition, and stereo vision, among many others. The purpose of this paper is to present a corner detection method for scene recognition in autonomous navigation of mobile robots. The proposed method is based on thirty windows to find possible corners. Some of the proposed windows are taken from fingerprint recognition papers and other are derived in this paper. The set of windows are used through a matching process to detect corners presented in the image under analysis. Results of the proposed method yield better quantitative and qualitative results compared with the Harris and SUSAN methods. These results can be used by a pattern recognition system for scene recognition.

Keywords: Scene recognition, corner detection, robotics.

1 Introduction

Corner detection is used in areas such as robotics, medicine. In robotics, it is used for data fusion, navigation, and scene recognition [1]. In medicine, it is applied for image registration such as x-rays, ultrasounds, and medical diagnostics [2]. In other applications, corner detection is used for object recognition, stereo vision, motion detection, among many other usages [3] [4].

Corners are the features more abundant in images of the real world, in contrast to straight lines [5]. For this reason, the use of corners is commonly found in tasks such as image matching. One of the advantages that corners offer is that, if we have images of the same scene, although taken from different perspectives, we will find almost the same corners, which is a good feature for image registration. That in turn will provide information for navigation of mobile robots. There are different definitions for what it is considered a 'corner'. F. Mokhtarian and R. Suomela, in [6], explain that the points of corners in an image are defined as points where the contour of the image has its maximum curvature. Juan Andrade-Cetto, in [7], mentions that corners are one of the most simple features which can be extracted from an image, and define corners as those points in an

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 11-19

image where there is change in the intensity in more than one direction. Krishnan Rangarajan, et al. [8], describe a corner as the point of union of two or more straight lines. In this paper, a corner is considered accordingly to the definition of Rangarajan. Based on this consideration, the proposed method described in this paper locates corners based on corner window analysis.

The paper is organized in the following sections. In Section 2, some of the more used methods for corner detection are discussed. Section 3 provides a general glance of the method proposed in this paper for corner detection. Sections 4 and 5 describe the main steps of the corner detection method. Results and conclusions are presented in Section 6.

2 Corner detection methods

The methods for the detection of corners can be divided in two groups: those which can accomplish the detection from the image in gray scale, and those which first detect edges and then detect corners. Among the methods of the first group, the most mentioned in the literature are the method of SUSAN [9] and the method of Harris [10]. The method of SUSAN differentiates from other methods in that it does not compute the derivative of the image under analysis and that it is not necessary to reduce the noise that could be present in the image. It uses a circular mask which scans the whole image, comparing the gray levels of the central pixel in the mask and the rest of the pixels inside the mask. All the pixels with a gray level equal to the central pixel level are considered as part of the same object. This area is called USAN (Univalue Segment Assimilating Nucleus). The USAN area has a maximum value when the center is in a plain region of the image, a mean value when it is on an edge, and a minimum value when it is on a corner.

The method of Harris is more sensitive to noise because it is based on the first derivative of the image. However, it is invariant to rotation, translation and illumination, which give it advantages over other methods. This method uses a window which scans the image and determine sudden changes in gray levels which results from rotating the window in several directions.

Among the second group of corner detectors, which use any method of edge detectors, we can mention the one of X.C. He and N.H.C. Yung [11]. They use the method of Canny and indicate the steps to follow for the detection of corners calculating the curvature for each edge.

Other authors use windows for corner detection from edge images, such as K. Rangarajan et al. [12]. In a similar way, G. Aguilar et al. [13] compare images of finger prints for the identification of persons using 3x3 windows. On those, they propose different bifurcations to be found, which we could call 'corners'. W. F. Leung et al., [14], use 23 windows of different bifurcations and 28 different windows of other type of corners for their detection in the finger print image using neural networks. The method described in this paper is based on the second group of corner detectors. Those which first apply edge detection and then detect corners using windows over the edge image. The next section explains the corner detection process developed in this paper.

3 General Description of the Corner Detection Method

The process for corner detection is shown in Figure 1. The original image $I(x,y)$ is convolved with a Gaussian filter G to remove noise that could be present in the image, yielding the image $I_s(x,y)$. A gradient operator and a threshold to determine the edges are applied to the image $I_s(x,y)$. These two operations correspond to the edge detection using the method of Canny. The Canny method was used because it yielded better results than the Sobel and other common edge operators. The resulting image $I_b(x,y)$ is convolved (*) with 30 corner detection windows, w_c of order 3x3, to detect the corners present in the image. The resulting image $I_e(x,y)$ contains the corners found.

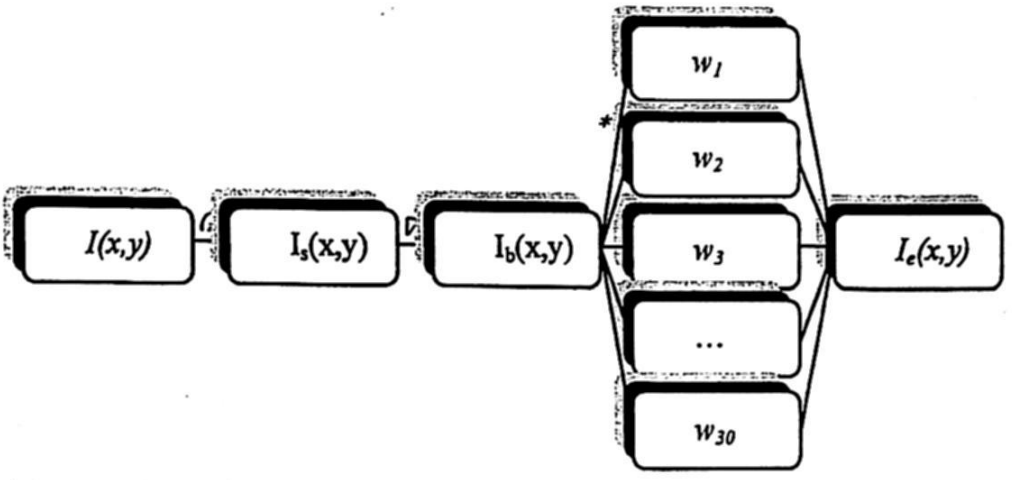


Fig. 1. Corner detection process.

4 Corner Detection Method

4.1 Corner detection windows

The papers from G. Aguilar [13], and W.F. Leung et al. [14], coincide in that there are different types of bifurcations or corners, that we call them, Y's, V's, T's, L's, and X's, accordingly to the form they take, as shown in Figure 2. Based on the similitude of these corners with fingerprint marks it was decided to investigate the possibility of using a unified theory between fingerprint recognition and scene recognition. Thus, from the finger print recognition works, some windows were chosen to detect corners. These selected windows plus other proposed in this paper make a set of 30 windows. Each corner detection window, w_c, is a 3X3 mask and some of their structures are illustrated in Figure 3.

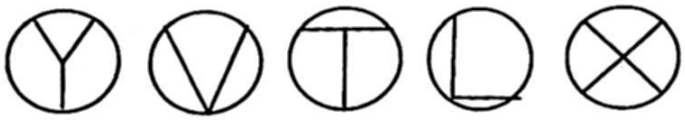


Fig. 2 Type of corners.

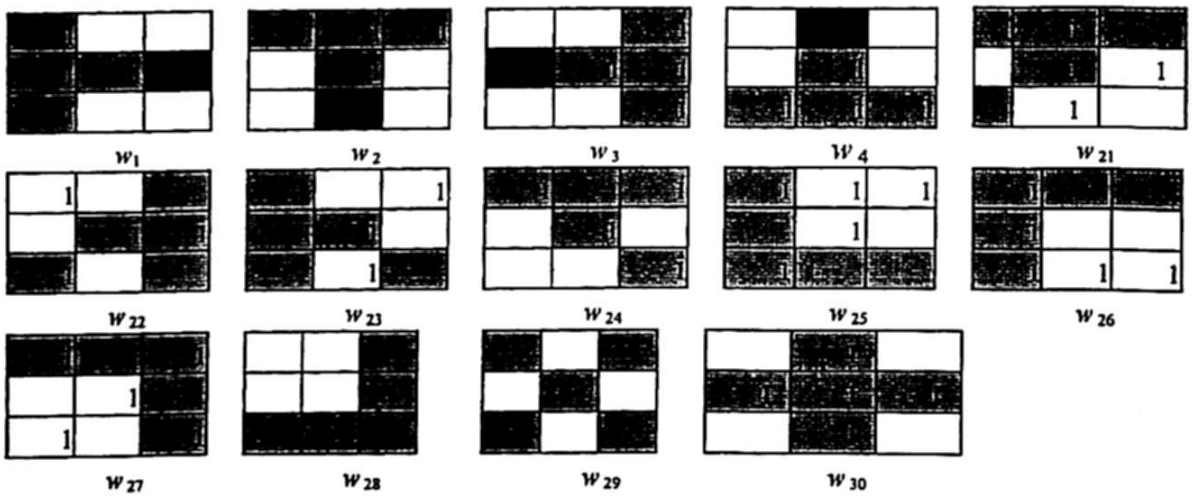


Fig. 3 Proposed corner detection windows.

The set of w_c windows is composed as follows. Windows w_1, w_2, w_3, and w_4, are four windows modified from the work of Leung at al. [14]. The modification consists on the aggregation of one pixel because they try to find terminal points and in our case we look for crossing lines. The extra pixel is darkened in these windows. Windows w_5 to w_{20} were also taken from Leung. The windows w_{17} to w_{20} appear in Aguilar et al. [13]. The subset w_{21} to w_{30} are windows proposed in this paper. The proposed windows were defined by analysis of the corners usually found in the set of images considered in this work.

4.2 Corner detection

Corner detection is achieved through a windows matching process. The image is scanned with the different corner detection windows w_c , trying to match the window corner shape with the edge pixels. Assuming a 3X3 neighborhood and two possible values $\{0,1\}$ for each pixel, the number of permutations is $2^9 = 512$. The values 2^n for n=0,1,..., 8 can be considered as weighting values to generate a generic weight matrix T_n , starting at $p(1,-1)$ as shown in Figure 4.

2^8	2^5	2^2
2^7	2^4	2^1
2^6	2^3	2^0

=

256	32	4
128	16	2
64	8	1

Fig. 4. Generic weight matrix, T_n.

Using the generic weight matrix T_n each w_c can be associated with an index window B_i

$$T_n : w_c \Rightarrow B_i \qquad for \quad c,i = 1,..,30 \tag{6}$$

obtained by

$$B_i = w_c \times T_n \ . \tag{7}$$

where the multiplication is an element by element multiplication and not a matrix multiplication. In this way, each w_c window is related to a index window B_i. In the same way, each index window B_i can be associated to a total weighting factor α_i obtained by equation (8).

$$\alpha_i = 1 + \sum_{b_i \in B_i} b_i \ . \tag{8}$$

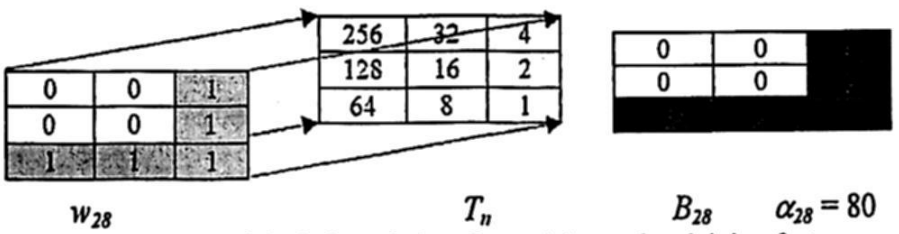
where the b_i correspond to the weighting factor in B_i. Figure 5 illustrates the computation of the index window B_{28} and the total weighting factor α_{28} of the window w_{28}. Once the main definition of the proposed methodology has been defined, the steps to detect the corners are now described.

Corner detection of a scene is accomplished by the next steps. First convolve the binary Canny result image $I_b(x,y)$ with the index matrix B_i

$$I_{ci}(x,y) = I_b(x,y) * B_i + 1 \ . \tag{9}$$

This step yields the possible corners related to each corner window w_c. The next step is to decide which of the possible candidate pixels in each $I_{ci}(x,y)$ is a corner that corresponds to w_c. This process is realized scanning the $I_{ci}(x,y)$ and assigning a pixel value according to

$$p_{ci}(x,y) = \begin{cases} 1 & p_{ci}(x,y) = \alpha_i \\ 0 & \text{otherwise} \end{cases} \tag{10}$$



w_{28} T_n B_{28} $\alpha_{28} = 80$

Fig. 5. Computation of the index window B_{28} and the total weighting factor α_{28} of w_{28}.

to produce a new set of images $I_{ei}(x,y)$ where $p_{ei}(x,y) \in I_{ei}(x,y)$ and $p_{ei}(x,y) \in I_{ei}(x,y)$. The value 1 indicates that the pixel $p_{ei}(x,y)$ is a corner of the type w_e. This process ends up with 30 binary images that indicate the position of the different type of corners. The final step consists on the union of the $I_{ei}(x,y)$images to produce the final corners

$$I_{FC}(x,y)=\bigcup_{i=1}^{30} I_{ei}(x,y) \quad . \tag{11}$$

5 Results

The proposed method was tested in semi-artificial scenes as well as with a set of real scenes. The purpose of using semi-artificial scenes is to obtain a correct performance measure of the proposed method, which will be hard to compute in real (noisy) scenes. These kind of images allows to compute false positives and false negatives detections in a simpler form and to achieve an analysis of these cases. A semi-artificial image is a simplified image extracted from a real scene. Figure 6 shows the semi-artificial version of three scenes.

Results of the application of the proposed method to the scenarios are shown in Figure 7. The first figure indicates the corners to be detected and the second the detected corners by the proposed method. There are 32 corners to detect in scene 1. The method detects 36 corners, therefore there are 4 false positive detections. The scene 2 contains 42 corners. The method found 40 corners, and there are 2 false negatives corners. In the scene 3 the method detected 61corners, 6 of them are false positives. These results yield 98.45% of real corner detections, 7.75% of false positive detection, and 1.56% of false negatives. The results of the Harris method using the same scenes are 96.9% of real corner detections, 33.33% of false positive, and 3.1% of false negative. Comparison with the SUSAN algorithm is not possible because it requires gray level information. A qualitative comparison will be given over the original images later on.

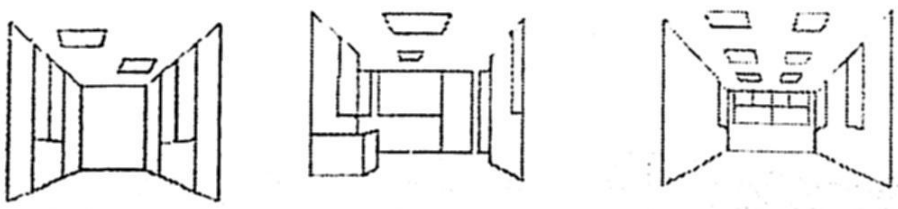


Fig. 6. Semi-artificial scenes, 1, 2 and 3.

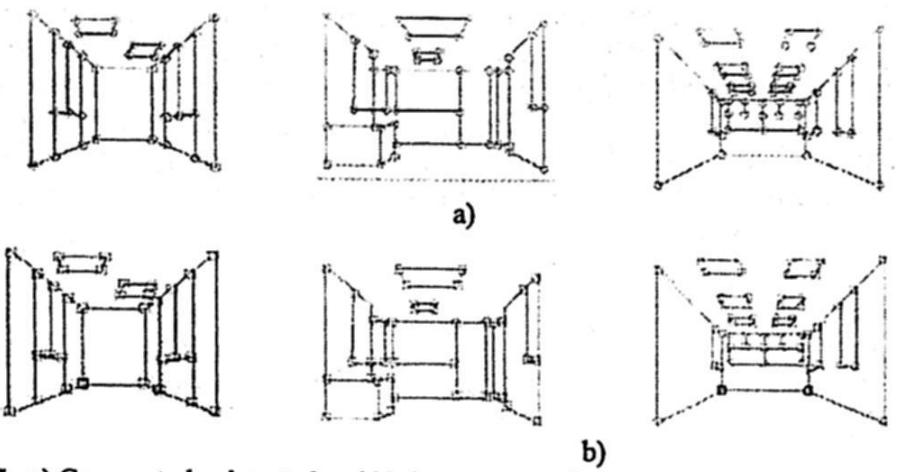


Fig. 7. a) Corners to be detected and b) the corresponding detected corners by the proposed method for scenes, 1, 2 and 3.

False negatives are mainly due to corners that do not match exactly to any w_c. False positives tend to appear as a consequence of multiple corner detections. This is because more than one w_c make a match with the image edge structure close to the corner. Figure 8 shows two multiple detection due in scene 1. Figure 8b illustrates the corner structures that make a match with the w_5, and w_8, windows.

The final results commented in this work are related to the comparison of the proposed method with the Harris and SUSAN methods applied in real scenes. Figure 9 show the corners detected by Harris (a), SUSAN (b) and the proposed method (c).

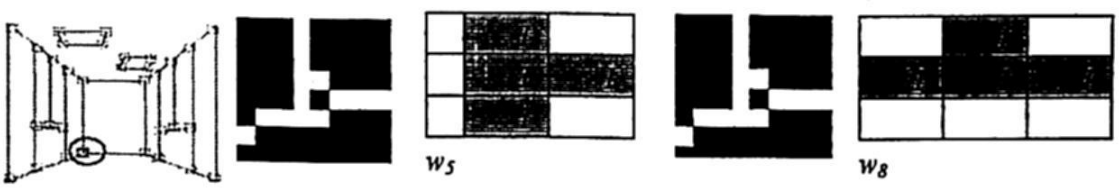


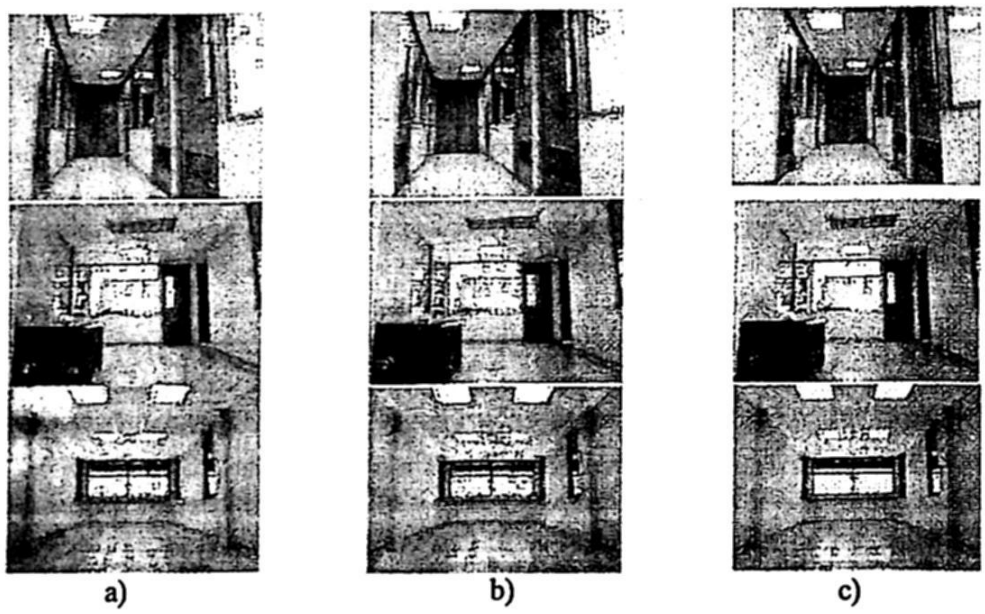
Fig. 8. False positives in scene 1.

It can be observed that, in general, Harris and SUSAN tend to detect more corners than the proposed method. However, the false positive rate is assumed to be very high, as proved with the semi-artificial images using the Harris method. Considering that corner information is used for robot navigation, high rate on false positives may lead to complicate more the scene recognition than the lack of some corners.

6 Conclusions

The proposed method has a better quantitative performance on semi-artificial scenes than the Harris method; and a better qualitative performance, with real scene images, than

the Harris and SUSAN methods. Scene recognition using features obtained from the detected corners presents preliminary acceptable results in a neural network scene recognition system like in [15]. The performance of this system is around 85% of correct recognition considering 10 different scenes considering different distance and orientations, four of them very alike, which makes the recognition harder. Future work includes more feature extraction experimentation over the detected corners, and the design of new scene recognition systems.



a) b) c)

Fig. 9. Corner detection by a) Harris b) SUSAN and c) Proposed method.

Acknowledgement

This work was supported by SEP-DGEST under Grants 2173.09-P and 2172.09-P.

7 References

[1] Se, S., Lowe, D., Little, J.: Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. International Journal of Robotics Research, pp. 735–758. (2002).

[2] Antoine Maintz, J. B., Max A. Viergever.: A Survey of Medical Image Registration. Medical Image Analysis, vol. 2, number 1, pp. 1–37, Oxford University Press (1998).

[3] Zitova, B., Flusser, J.: Image Registration Methods: a survey. Image and Vision Computing 21, pp. 977–1000, (2003).

[4] Bradski, G., Kaehler, A.: Robot-Vision Signal Processing Primitives. IEEE Signal Processing Magazine, pp. 130- 133, January (2008).

[5] Tissainayagam, P., Suter, D. .: Assessing the Performance of Corner Detectors for Point Feature Tracking Applications. Image and Vision Computing, vol. 22, pp. 663-679, Elsevier (2004).

[6] Mokhtarian, F., Suomela, R. .: Curvature Scale Space for Robust Image Corner Detection. Int'l Conf. Pattern Recognition, Brisbane, Australia (1998).

[7] Andrade, J.: Environment Learning for Indoor Movile Robots. PhD Thesis, Universidad politecnica de Catalunya, (2003).

[8] Rangarajan, K., Shah, M., van Brackle, D.: Optimal Corner Detector. Second International Conference on Computer Vision, pp. 90 – 94, December (1988).

[9] Smith, S.M., Brady, J.M.: SUSAN - A New Approach to Low Level Image Processing. Int. Journal of Computer Vision, vol. 23 (1), pp.45-78, May (1997).

[10] Harris, C.G., Stephens, M.: A combined corner and edge detector. In proceedings of the Alvey Vision Conference, pp. 189-192. Manchester (1988)

[11] He, X.C., Yung, N.H.C.: Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region of Support. Proceedings of the 17th International Conference on Pattern Recognition, vol. 2, pp: 791- 794. Aug. (2004).

[12] Zitová B., Flusser J., Kautsky J., Peters G.: Feature Point Detection in Multiframe Images. Proceedings of the Czech Pattern Recognition Workshop (Svoboda T. ed.), Czech Pattern Recognition Society, pp. 117-122, Praha (2000).

[13] Aguilar, G., Sanchez, G., Toscano, K., Salinas, M., Nakano, M., Perez, H.: Fingerprint Recognition. Second International Conference on Internet Monitoring and Protection, ICIMP 2007, pp. 32 – 32, July (2007).

[14] Leung, W.F., Leung, S.H., Lau, W.H., Luk, A.: Fingerprint Recognition Using Neural Network, Neural Networks for Signal Processing [1991], Proceedings of the 1991 IEEE Workshop, pp. 226 – 235, 30 Sept.-1 Oct. (1991).

[15] Wang, S., Lei, S., Chang, F.: Image registration Based on Neural Network. Proceedings of the 5th , International Conference on Information Technology and Application in Biomedicine, in conjunction with The 2nd International Symposium & Summer School on Biomedical and Health Engineering, pp 74-77, Shenzhen, China, May (2008).

Incorporating Angular Ratio Images into Two-Frame Stereo Algorithms

Pablo-Arturo Martínez-González and Mario Castelán*

Centro de Investigación y de Estudios Avanzados del I.P.N.
Robotics and Advanced Manufacturing Group,
Ramos Arizpe, Coahuila, 25900, México

Abstract. The Light Transport Constancy (LTC) establishes that the reflectance ratio obtained from two different illumination variations remains constant for any given view of the observed scene. In [1], LTC was proposed as a rank constraint for solving the correspondence problem in multiple view stereo. In two-frame stereo, the simplest setting for LTC requires only two illumination variations and a single light source. Under this scenario, the rank constraint can be formulated through ratio images and standard stereo algorithms be applied in order to obtain a disparity map. Unfortunately, a ratio image may be subject to saturated pixel values, which may diminish the quality of the disparity maps. To overcome this problem, we introduce a post-processing operation based on the slope angles related to the ratio values. Experiments show that the new angular ratio images are more robust and deliver improved disparity maps.

Key words: Light Transport Constancy, two-frame stereo, ratio images.

1 Introduction

Acquiring the three dimensional surface of objects is an important problem in Computer Vision, since a 3D surface simplifies the modeling of the appearance of the object. A 3D surface can be obtained using contact devices such as laser scanners. Other possibilities imply information provided by one or more cameras. This methodology is known as image-based 3D shape recovery. Although the image-based approach is appealing, the nature of the image acquisition process makes the input images prone to errors. In this sense, lighting manipulation represents a way to posing constraints on image-based shape recovery techniques. For instance, the intensity of the light can be regulated in order to obtain 3D shape. This is the core idea of Light Fall-off Stereo (LFS)[7], where a number of images is gathered from a stationary camera as the illumination source moves away from the scene. Based on the inverse square law for light intensity, the ratio images are directly related to scene depth from the perspective of the light source. Controlling the geometric position of the light source represents a

* This work has been supported by Project Conacyt Ciencia Bsica 61593.

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 21-28

different alternative to face the problem. The Photometric Stereo Method (PSM) [6] is the classical technique dealing with this consideration. Here, a single camera captures images while the direction of a light source moves around a fixed pose object.

When more than one camera is required, binocular stereo (two-frame stereo) is the image-based 3D shape recovery method with the simplest setting. Here, only two cameras capturing a still scene are needed and the correspondence problem is solved between the two views in order to obtain depth information. Unfortunately, when Lambertian reflectance and color/brightness constancy are not observed, the calculation of correspondences becomes a difficult task. In binocular stereo, the manipulation of lighting has also been proposed. For example, projective geometry laws may be applied if structured light patterns are projected over the surface of the object [5]. This is normally done using a projector, but colored laser rays can also be projected if more accurate results are needed. Another approach based on lighting variations is the Helmholtz stereopsis. This method allows matching of arbitrary Bidirectional Reflectance Distribution Functions (BRDF) and uses the reflectance function reciprocity as an invariant [8, 9] . By collocating point light sources with each camera it is possible to record reciprocal pairs using two different lighting conditions. Due to reciprocity the reflected light to the cameras will be equal. This method, however, requires the light sources to be collocated with the optical center of each camera.

Recently, the Light Transport Constancy (LTC) [1] has been proposed as a correspondence clue in multiple-view stereo. LTC is used to formulate a rank constraint matching cost when the scene is observed in several lighting variations (i.e., changes in light intensity). LTC establishes that the reflectance ratio obtained from two different illumination variations remains constant for any given view of the observed scene. LTC does not require the position of light sources to be precisely calibrated or even known. In two-frame stereo, the simplest setting requires only two illumination variations. Under this scenario, the rank constraint can be formulated through ratio images, and standard stereo algorithms be applied in order to obtain a disparity map. Unfortunately, a ratio image may be subject to saturated pixel values, noise and occlusions, which may diminish the quality of the disparity maps. To overcome this problem, in this paper, we introduce a post-processing operation based on the slope angles related to the ratio values.

The paper is organized as follows: in Section 2, the LTC is introduced; in Section 3, the use of angular measures for improving ratio images is explained; experiments are presented in Section 4 and finally conclusions are described in Section 5.

2 Light transport constancy

The Light Transport Constancy states that the percentage of light reflected by a surface patch (the BRDF) remains constant for any given viewing direction of

a static scene. Following the explanation in [1], let us call a particular point in the scene x_i. This point will reflect light to the cameras C_1 and C_2 according to $I_{C_j}(x_i) = L(x_i)R(x_i, L, C_j)$, where $I_{C_j}(x_i)$ is the reflected intensity in the direction of C_j from the point x_i, $L(x_i)$ is the incident light intensity at point x_i, and $R(x_i, L, C_j)$ is the reflectance function or BRDF at point x_i, indexed by the vectors in the direction of L and C_j. The Lambertian assumption asserts that the reflected light is equal in the directions of C_1 and C_2, i.e., the BRDF is shared and $R(x_i, L, C_1) = R(x_i, L, C_2)$. Thus we have $I_{C_1}(x_i) = I_{C_2}(x_i)$. However this relation will not in general hold for arbitrary (non-Lambertian) BRDFs. Light transport constancy assumes that the surface reflectance function, $R(x_i, L, C_j)$, remains constant under variable illumination. If we vary the lighting conditions, so that the incident illumination varies by a factor of $k(x_i)$, then the observed reflected light, $I_{C_j}(x_i)$, will also vary by a factor of $k(x_i)$, as

$$I_{C_j}(x_i) = k(x_i)L(x_i)R(x_i, L, C_j) \tag{1}$$

Wang et al [1] have shown how LTC can be used in multiple-view stereo to impose a rank constraint on the matrix

$$I_{CV} = \begin{pmatrix} I_{C_1V_1} & I_{C_2V_1} & \cdots & I_{C_mV_1} \\ I_{C_1V_2} & I_{C_2V_2} & \cdots & I_{C_mV_2} \\ \vdots & \vdots & \ddots & \vdots \\ I_{C_1V_n} & I_{C_2V_n} & \cdots & I_{C_mV_n} \end{pmatrix}, \tag{2}$$

where $I_{C_jV_k}$ is the observed grayscale value by the j_{th} camera under the k_{th} lighting variation. Note that, for the sake of simplicity, we have omitted the notation (x_i). However, each of the remaining equations in the paper is related to a single pixel at position x_i. The matrix with minimum rank is therefore sought, i.e., if LTC is observed through the different camera viewing positions and lighting variations, then the dimension of the column space of the matrix I_{CV} should be minimal. The rank constraint only holds when the number of light sources is less than both the number of lighting variations and the number of cameras. Then the rank of I_{CV} is at most the number of light sources. Since I_{CV} will be corrupted with noise, it is impossible to calculate rank exactly. The Singular Value Decomposition of I_{CV} may be used for rank approximation. A matrix with most of their energy in the first few principal components is preferred and moments can be used to approximate the notion of minimum rank, as

$$\mathcal{M} = \sum_i i \cdot \sigma_i^2 \Big/ \sum_i \sigma_i^2, \tag{3}$$

where σ_i are the singular values of I_{CV}. For multiple-view and multiple-lighting stereo, the minimum score is used as matching cost.

3 Improved ratio Images and the two-frame case

Let us now consider the simplest setting for LTC-based stereo: a single light source and two cameras. For each pixel in the left and right images, the intensities

observed by the first lighting variation can be explained in terms of the intensities observed by the second lighting variation, as $I_{C_1V_2} = I_{C_1V_1}k_1$ and $I_{C_2V_2} = I_{C_2V_1}k_2$. Therefore the relation between lighting variations is given by the ratio:

$$\frac{I_{C_1V_1}}{I_{C_1V_2}} = \frac{I_{C_1V_1}}{I_{C_1V_1}k_1} = k_1, \tag{4}$$

$$\frac{I_{C_2V_1}}{I_{C_2V_2}} = \frac{I_{C_2V_1}}{I_{C_2V_1}k_2} = k_2. \tag{5}$$

The matrix of intensities, I_{CV} can be now defined as

$$I_{CV} = \begin{pmatrix} I_{C_1V_1} , I_{C_2V_1} \\ I_{C_1V_2} , I_{C_2V_2} \end{pmatrix} \tag{6}$$

Note that LTC only holds if the second singular value of I_{CV} is zero. This means that the minimum rank of I_{CV} is one (the number of light sources) if and only if $k_1 = k_2$. Minimizing the second singular value is therefore equivalent to minimizing Eq. 3. The so-called *ratio image* is a function $R(v_1, v_2, x_i) = k_i$, where v_1 and v_2 are two different lighting variations. The ratio image is only defined for the two-frame/two lighting variations scenario and at most two ratio images can be recorded for a given stereo image pair, as in Eq. 4 and Eq. 5. Note how the minimization of Eq. 3 can also be carried out using a simple absolute difference matching cost over the ratio image pair. In this sense, a wide variety of two-frame stereo algorithms [2] provides many ways to calculate dense disparity maps through ratio images.

Unlike the classical grayscale (or color) image pair which usually assumes brightness/color constancy, the ratio images rely on the light transport constancy and have proved to provide improved disparity maps [1]. For these reasons, Wang et al. have pointed out the potential use of ratio images in the future as the most feasible and robust way to approach the two-frame stereo problem. Using ratio images for such a task may borrow ideas from the vast literature related to two-frame stereo algorithms [2, 4, 3]. For example, from the the taxonomy of Scharstein and Szelisky [2], different matching costs, aggregation support and optimization approaches can be applied to add robustness in the calculation of dense disparity maps from ratio images.

In practice, unfortunately, the intensities of the pixels do not necessarily observe $k_1 = k_2$. This is due to several reasons, among which insufficient lighting and a poor camera response are the most common. Moreover, in some regions of the image, i.e., where specularities and edges occur, the ratio is likely to be either a value close to zero or an overly saturated value, i.e., much greater than 100%. The unwanted effect of these pixel values can be reduced if the ratios are redefined as:

$$k_1' = |\tan^{-1}(k_1)|, \qquad k_2' = |\tan^{-1}(k_2).|, \tag{7}$$

where $|\cdot|$ is the absolute value. The angular data constrain the ratio values from the interval $[0, \infty)$ to the interval $[0°, 90°]$. We can now define the new ratio as $R'(v_1, v_2, x_i) = k_i'$ which will be referred to as the *angular ratio image*.

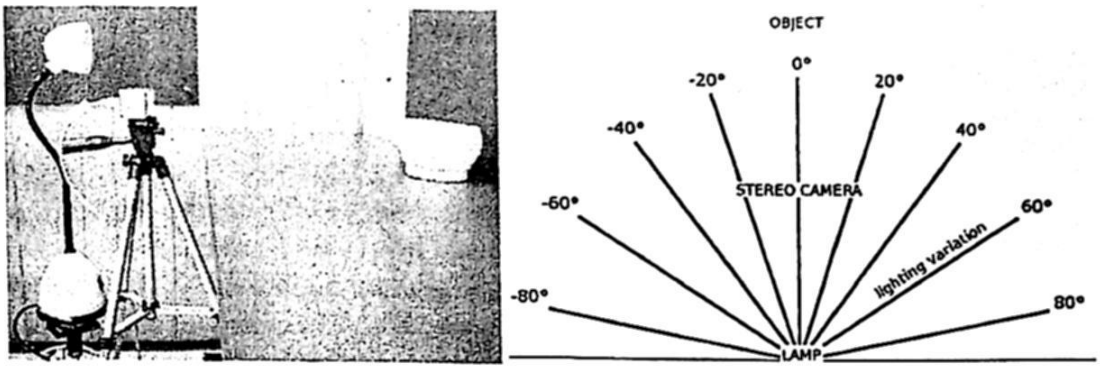


Fig. 1. The general setting and the illumination variation. The left image shows the general acquisition setting: the stereo camera, the desk lamp and the object. The lamp was mounted on a rotating ruler on a tripod, so that rotations could be measured. The right image shows a top-down sketch of the setting, where the degree of separation between each camera shot can be seen.

4 Experiments

For image acquisition, a Bumblebee stereo camera, $9cm$ baseline was used. The size of the grayscale images was 640×480 pixels. A halogen-bulb desk lamp was mounted on a rotating ruler in order to capture illumination variations around a range of 180^o, with 20^o increment as shown in Figure 1 (right). A picture of the general setting is shown in Figure 1 (left).

Before the ratio values are converted into image values (grayscale) a normalization operation has to be performed. Let $R(v_1, v_2, x_i)$ be the ratio values obtained from a pair of left images with lighting variations (v_1, v_2) (the same observations holds for its corresponding right pair), i.e., using Eqs. 4 and 5. A ratio image is stored after the following filter is applied:

$$R(v_1, v_2, x_i) = \begin{cases} \tau & \text{if } k_i \geq \tau \\ k_i & \text{otherwise} \end{cases} \tag{8}$$

where τ is a cut ratio value. Once the high values have been filtered, the final image is generated from the normalized values:

$$R(v_1, v_2, x_i) = \frac{R(v_1, v_2, x_i)}{\tau}. \tag{9}$$

Note that, once Eq. 8 is applied, τ becomes the maximum value of $R(v_1, v_2, x_i)$. As far as angular ratio images are concerned, i.e. $R'(v_1, v_2, x_i)$, Eq. 8 is not required, since Eq. 9 can be directly applied with $\tau = \pi/2$ (radians). Once a ratio (or angular ratio) image is generated, dense disparity maps are calculated using the standard platform developed in [1]. The *sum of absolute differences* (SAD) and the mean filter 9×9 window were respectively used as matching cost and aggregation support parameters. The disparity maps were finally calculated under the *winner-takes-all* (WTA) criteria.

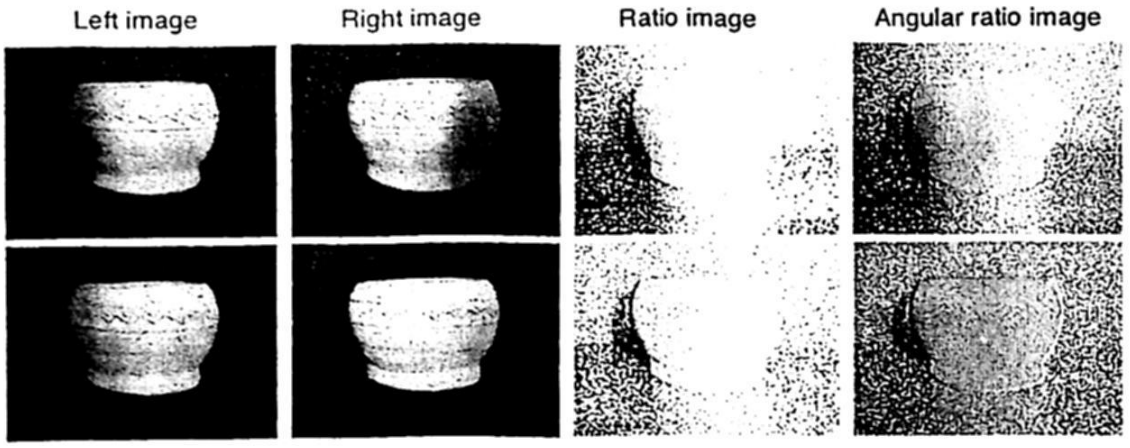


Fig. 2. Ratio images and angular ratio images. From top to bottom, the different rows of the figure show the lighting variation pairs $(-60°, 60°)$ and $(-40°, 40°)$. The left-camera image, right-camera image, ratio image and angular ratio image are shown row-wise.

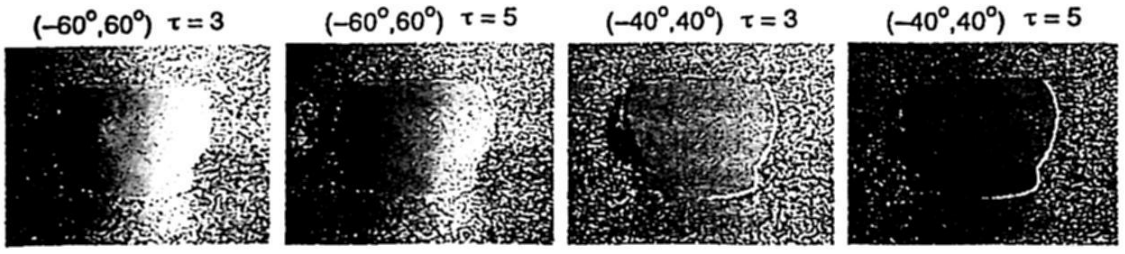


Fig. 3. Varying τ for ratio image generation. Normalization results for $\tau = 3$ and $\tau = 5$ are shown with lighting variations as depicted in the labels of the figure.

The experimental analysis commences with Figure 2, where the difference between ratio and angular ratio images can be appreciated. From top to bottom, the different rows of the figure show the lighting variation pairs $(-60°, 60°)$ and $(-40°, 40°)$. The left-camera image, right-camera image, ratio image ($\tau = 1$) and angular ratio image are shown row-wise. Recall that the ratio and angular ratio images are calculated for a single view (i.e., left or right camera images) and two variations. From the figure, it is noticeable that the angular ratio images reveal a more robust adjustment of values than the ratio images, where the cut value has set the ratios to saturated values. This is a main problem in ratio image generation, i.e., choosing an optimum cut value. This effect can be visualized in Figure 3, where different values of τ are applied to different lighting variations. Again, a generalized optimum value of τ is not clear, as the figure shows that $\tau = 5$ favors the ratio obtained from the pair $(-60°, 60°)$, but over-darkens the ratio obtained from the pair $(-40°, 40°)$.

As far as disparity results are concerned, these are depicted in Figure 4. Here, the first row presents disparity maps while the second row shows a line of interest with constant $y - axis$ along the disparity map. For visualization purposes, a

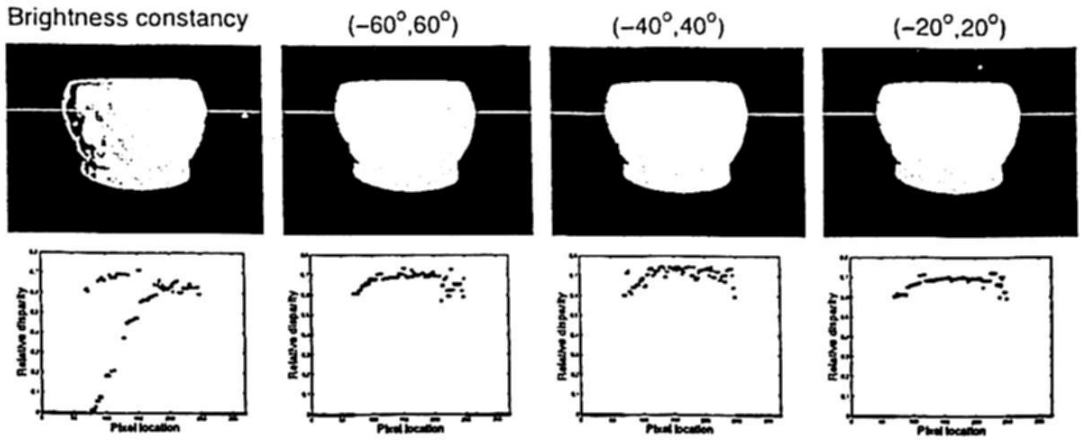


Fig. 4. Disparity results on brightness constancy and angular ratio images. The first row shows disparity maps, the second row shows a line of interest with constant $y - axis$ along the disparity map.

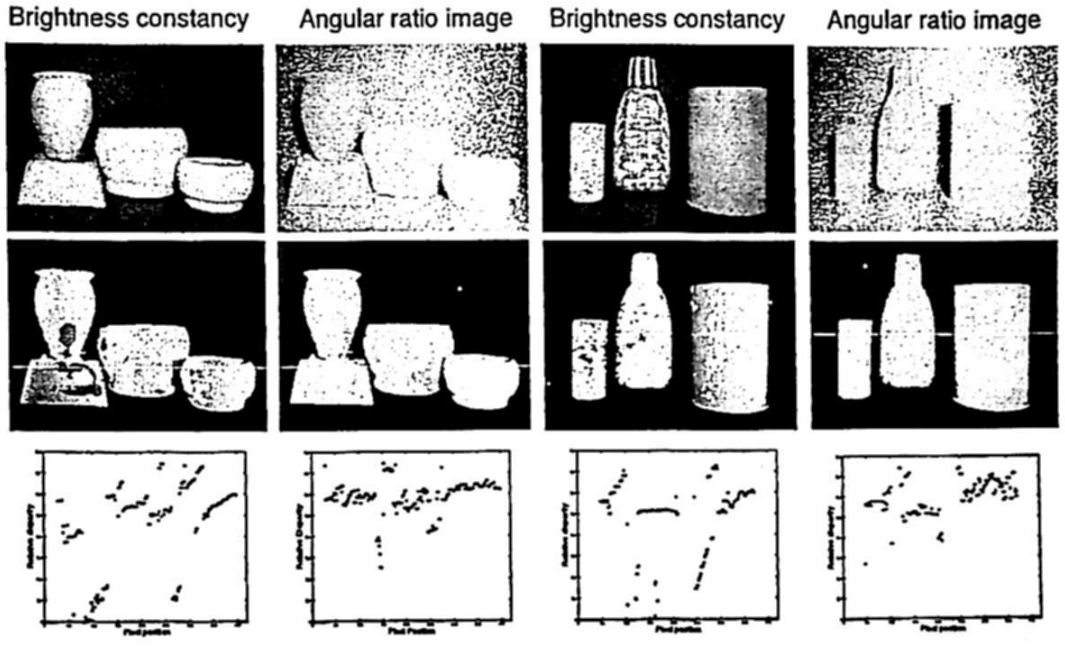


Fig. 5. Additional scenes. The first row presents the kind of input images used, the second row presents the obtained disparity maps. The third row presents a line of interest.

mask has been applied for isolating the objects of interest from the background. The first row presents results for brightness constancy, i.e., neither ratio nor angular ratio images are used here. Instead, the usual grayscale image left/right image pair is used as stereo input. The rest of the columns present results from angular ratio images with lighting variation pairs of $(-60^o, 60^o)$, $(-40^o, 40^o)$ and $(-20^o, 20^o)$. As expected, there is an improvement in disparity calculation for the angular ratio images over the brightness constancy, i.e., the disparities are

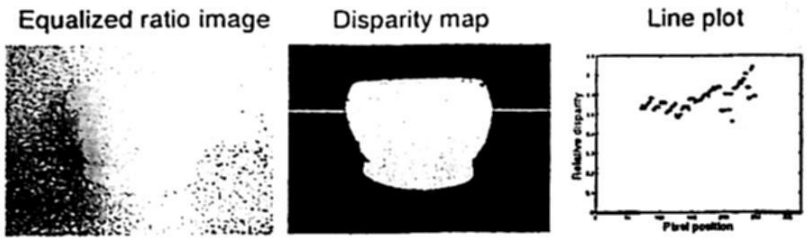


Fig. 6. Results on a histogram-equalized ratio image. From left to right: equalized ratio image, disparity map, line of interest along the disparity map.

located over more continuos regions. Interestingly, there is little difference between the results related to angular ratio images, which suggests the angles may represent a robust way to obtain similar disparity maps through the different lighting variations. Additional scenarios are presented in Figure 5, again, the disparity results for the angular ratio images outperform those relying on the brightness constancy assumption.Finally, we explore using histogram equalization as an alternative adjustment function for ratio images and the outcome of the experiment is shown in Figure 6. Here, a standard histogram equalization operation was performed on the raw ratio image with variation $(-60°, 60°)$. Despite the equalized ratio image look similar to its angular ratio image counterpart (top right corner of Figure 2), the disparity results are rather different, favoring again the use of angular ratio images.

5 Conclusions

A new way to define ratio images for LTC two-frame stereo has been presented. The new angular ratio image does not require cut ratio values and suggests to be robust to illumination variations. This facilitates experimenting with known two-frame stereo algorithms under the LTC stereo setting. As future work, we plan to study different aggregation and optimization methods on these images.

References

1. L. Wang, R. Yang, and J. E. Davis, BRDF Invariant Stereo Using Light Transport Constancy. *IEEE Trans. Pattern Anal. and Mach. Intell.*, Vol. 29, No. 9, pp. 1616–1626, (2007).
2. D. Scharstein and R. Szelisky. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *Int. Journal of Comp. Vis.*, 47(1):7–42, 2002.
3. H. Hirschmüller and D. Scharstein. Evaluation of Stereo Matching Costs on Images with Radiometric Differences. *IEEE Trans. Pattern Anal. and Mach. Intel.*, 31(9):582–1599, 2009.
4. M. Gong, R. Yang, L. Wang and M. Gong. A Performance Study on Different Cost Aggregation Approaches Used in Real-Time Stereo Matching. *Int. Journal of Comp. Vis.*, 75(2):283–296, 2007.
5. D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. *Proc. IEEE ICCV*, 2003.
6. R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Enginnering* 19(1):139–144, 1980.
7. M. Liao, L. Wang, R. Yang and M. Gong. Light Fall-off Stereo. *Proc. IEEE CVPR* 1–8, 2007.
8. S. Magda, D. J. Kriegman, T. Zickler, and P. N. Belhumeur. Beyond Lambert: Reconstructing surfaces with arbitrary BRDFs. *Proc. IEEE ICCV*, 2, pp. 391, 2001.
9. T. E. Zickler, P. N. Belhumeur, and D. J. Kriegman. Helmholtz stereopsis: exploiting reciprocity for surface reconstruction. *Int. Journal of Comp. Vis.*, 49:215, 2002.

Relative Depth Estimation of Objects in Underwater Scenes

Ricardo Pérez-Alcocer, L. Abril Torres-Méndez and Ernesto Olguín-Díaz

Robotics and Advanced Manufacturing Group, CINVESTAV-Saltillo
Ramos Arizpe, Coahuila, MEXICO, 25900

Abstract. A method to estimate the relative depth of objects in an underwater environment is presented. The method uses as a depth cue the color of the objects without considering the degradation produced by the scattering and attenuation factors. To establish a relationship of the color intensity values, two images of the objects under different conditions are needed. In our case, one image is taken outside the water and the other underwater. For the preliminary experiments, we have generated underwater images from non-aquatic images with known range maps. We then apply our model to estimate the relative depth of the simulated underwater image. To evaluate the performance of our method, we first normalize both the resulting range map and ground truth range map and compared them.

1 Introduction

As underwater applications are becoming more feasible by current technology, the connection between depth perception and the actions to be taken in this kind of environments is fundamental. A great number of research works have focused on the development of underwater devices that autonomously can recollect, analyze and interpret information from this type of ecosystems. In this trend, the use of vision systems for aquatic robots ([1], [2], [3]) has become crucial for many underwater inspection and observation tasks. Unfortunately, the physical properties of underwater environments impose severe limitations on the quality of optical imaging due to the attenuation and scattering of light; and the living organisms in water. Furthermore, the time of the day and cloudiness of the sky have a great effect on the nature of the light available. Ambient light is practically nonexistent after few meters of depth due to the rapid attenuation of electromagnetic radiation underwater. The light undergoes scattering along the line of sight. The result is an image that is color depleted, blurry and out of focus. By 3m in depth there is almost no red light left from the sun. By 5m, orange light is gone, by 10m most yellow is also gone. By the time one reaches 25m only blue light remains [4]. Since many of the above factors are constantly changing, we cannot really know all the effects of water.

Some research works have proposed different ways to model the physical interaction of light in water [5], [4], [7]. However, if we want to have a more realistic

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 29-36

model, we need to take into account other factors, such as all possible interactions of light sources with the objects; the analysis of the chemical components present in water depending on its source (e.g., lakes, rivers, ocean or even artificial pools). Clearly, we are in front of a very complex situation, as many different parameters (whose values are typically empirically assigned) need to be considered. In order to avoid this, it is common to assume aspects about the patterns present in the images to be analyzed and also about the environment on which they were captured. In this trend, some methods try to enhance visibility by decreasing the optical effects of scattering [6]. Another related application of such models is the color correction of images [7], [8]. In general, by understanding these models, we can be able to artificially generate images having an aquatic appearance from images taken on the surface. This is useful to perform analysis and recognition tasks on the patterns in the images and be able to estimate depth information, which is our main interest.

In this work, color information is used as a depth cue. First, we use the model proposed by Schechner and Karpel in [7] to artificially generate aquatic images. Once we have two images from the same scene, but under different conditions, we can estimate the relative depth information of each object in the underwater scene. To achieve this, we propose a depth estimation model based on the ideas presented in [6].

This paper is structured as follows. Section 2 presents the image formation model for aquatic images. In Section 3, we describe our method to estimate relative depth information from underwater images. Section 4 tests the proposed method on different generated underwater images. Finally, in Section 5 we give some conclusions and future directions.

2 The Image Formation Model for Underwater Images

The first step in the development of this work is to implement a simple model of underwater image formation. The understanding of certain properties observed in the aquatic images obtained from this model will help us to define an efficient method for depth estimation of underwater scenes.

In the formation of underwater images two main sources of light are present; the radiance of the objects in the environment and the ambient light. They both give as a result the characteristic tones existing in photos taken underwater.

In general, the signal is composed by two components denominated *direct transmission* and *forward scattering*. During the propagation of light rays coming from the objects to the camera, part of the energy is lost due to the scattering and the absortion. The component that represents these two factors is called direct transmission and is given by

$$D = L_{object} e^{-\beta z} \tag{1}$$

where β is the attenuation coefficient and L_{object} is the radiance of the object.

The component of forward scattering is similar to the direct transmission component. However, the forward scattering component generates blurred images given by the following convolution

$$F = D * g_z \tag{2}$$

where D is given by Eq. (1) and g_z is a point scattering function (PSF). The PSF is parametrized by the distance z. For this reason, objects located at far distances look blurrier than near objects. An example of a PSF is given by

$$g_z = (e^{-\gamma z} - e^{-\beta z})\mathcal{F}^{-1}\{G_z\} \quad \text{and} \quad \mathrm{G_z} = \mathrm{e}^{-\mathrm{Kz}\omega} \tag{3}$$

where $K > 0$ and γ is a heuristic constant, $\mathcal{F}^{-1}$ is the Fourier inverse transform, and ω is the spatial frequency in the image plane.

Taking into account the direct transmission and the forward scattering, the signal can be defined as $S = D + F$. And defining $L_{object}^{effective}$ as

$$L_{object}^{effective} = L_{object} + L_{object} * g_z \tag{4}$$

we obtain that the signal is given by

$$S = e^{-\beta z} L_{object}^{effective} \tag{5}$$

On the other hand, the second component of the model is given by the backscattering that comes from the ambient illumination, from [7], the total backscattering is given by

$$B_\infty(1 - e^{-\beta z}) \tag{6}$$

Therefore, taking into account the signal and the backscattering, the total irradiance of the image is given by

$$I^{total} = S + B = e^{-\beta z} L_{object}^{effective} + B_\infty(1 - e^{-\beta z}) \tag{7}$$

Using the model described by Eq. (7) we can obtain simulated aquatic images by using an image taken on the surface and its associated range map. We have implemented this model and show some examples. We use the Middlebury[1] dataset in all our experiments. The original intensity images are depicted in the left column of Figure 1 and in their right are the associated range images. Some examples of the resulting aquatic images are shown in Figure 2 after applying the model explained above with the following parameters: $B_\infty = [48, 250, 255]$ and $\beta = [0.4, 0.0991, 0.0348]$. It can be noted that the images take a blue tone which is characteristic of underwater images. Also, it can be noted the effect that far objects look blurrier than near objects.

We will use this image formation model to simulate aquatic images and be able to validate our depth estimation model, which is described next.

[1] http://cat.middlebury.edu/stereo/data.html

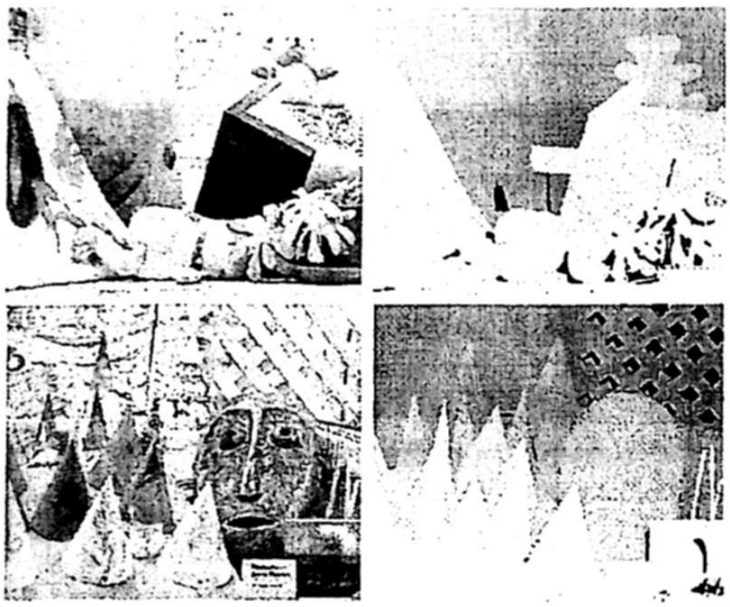


Fig. 1. Two examples of the original intensity images with their associate range images.

3 The Proposed Depth Estimation Model

Based on the comparative analysis carried out between the generated aquatic images and real aquatic images, some simplifications can be made to define our depth estimation model. The first simplification is that the forward scattering does not affect in a considerable manner. This is reasonable when the underwater illumination conditions also do not vary considerably in a given period of time. Thus, the Eq. 7 can be simplified for each of the RGB channels and is given by

$$\begin{aligned} E_R &= I_{OR}e^{-\beta_R d} + A_R(1 - e^{-\beta_R d}) \\ E_G &= I_{OG}e^{-\beta_G d} + A_G(1 - e^{-\beta_G d}) \\ E_B &= I_{OB}e^{-\beta_B d} + A_B(1 - e^{-\beta_B d}) \end{aligned} \tag{8}$$

where $I = [I_{OR}, I_{OG}, I_{OB}]$ represents the pixel values in the image taken outside the water, $\beta = [\beta_R, \beta_G, \beta_B]$ contains the degradation coefficients for each of the color channels. An important aspect to note here is that the three degradation coefficients are different, but they remain constant through all pixels in the image. On the other hand, $A = [A_R, A_G, A_B]$ contains the color values of the pixels that have an infinite distance to the camera. Therefore, considering the total radiance of the image in the red channel, we have

$$\begin{aligned} E_R &= A_R - (I_{OR} - A_R)e^{-\beta_R d} \\ E_R - A_R &= (I_{OR} - A_R)e^{-\beta_R d} \\ e^{-\beta_R d} &= \frac{E_R - A_R}{I_{OR} - A_R} \end{aligned} \tag{9}$$

Fig. 2. Simulated aquatic images after applying the image formation model to the images in Figure 1.

In similar way we can obtain the formulas for the green and blue channels:

$$e^{-\beta_G d} = \frac{E_G - A_G}{I_{OG} - A_G} \tag{10}$$

$$e^{-\beta_B d} = \frac{E_B - A_B}{I_{OB} - A_B} \tag{11}$$

By dividing Eq. 9 and Eq. 10 and, considering that the value of β for each color channel is different, we have that

$$R_1 = e^{-(\beta_R - \beta_G)d} = \frac{(E_R - A_R)(I_{OG} - A_G)}{(I_{OR} - A_R)(E_G - A_G)} \tag{12}$$

And applying the natural logarithm to Eq. 12:

$$\ln(R_1) = -(\beta_R - \beta_G)d = \ln\left[\frac{(E_R - A_R)(I_{OG} - A_G)}{(I_{OR} - A_R)(E_G - A_G)}\right] \tag{13}$$

Now, if we calculate $\ln(R_1)$ for two pixel points i, j of the image and compute their ratio, we can obtain the relative depth for the two points in the scene.

$$\frac{\ln(R_1^i)}{\ln(R_1^j)} = \frac{z_i}{z_j} \tag{14}$$

Therefore, the relative depths for all pixels in the aquatic image can be calculated by knowing the intensity values (in the three color channels) of the non-aquatic image. Given that the computation of Eq. 14 for one pixel can contain certain level of perturbation, we made the model more robust by calculating

$$\frac{\ln(R_1^i)}{\sum_{j=0}^{j=N} \ln(R_1^j)} = \frac{z_i}{\sum_{j=0}^{j=N} z_j} \tag{15}$$

where $\sum_{j=0}^{j=N} z_j$ is an arbitrary constant that represents the sum of the distances of all pixels in the image.

An important aspect of the depth estimation method is that it obtains three estimates for each pixel by calculating R_1 as the ratio between (9) and (10), between (9) and (11), and also between (10) and (11), thus we have that

$$R_1 = e^{-(\beta_R - \beta_B)d} = \frac{(E_R - A_R)(I_{OB} - A_B)}{(I_{OR} - A_R)(E_B - A_B)} \tag{16}$$

$$R_1 = e^{-(\beta_G - \beta_B)d} = \frac{(E_G - A_G)(I_{OB} - A_B)}{(I_{OG} - A_G)(E_B - A_B)} \tag{17}$$

4 Experimental Results

In this section, we present some of the experimental results obtained after using our depth estimation model. We carried out several tests on different images by assigning different values to the degradation coefficient and compare the results.

In Fig. 3 we show two different scenes with two different degradation coefficients each. The first and third images in the left column are the generated aquatic images after applying the image formation model with degradation coefficient $\beta = [0.4, 0.0991, 0.0348]$ to the original images in Fig. 1. These images correspond to Case I. Here, the maximum distance to an object in the image is 0.8m. The second and last images in the left column are the simulated images when using $\beta = [0.1, 0.05, 0.005]$ and corresponds to Case II. The middle column shows the estimated range maps for each of the image at the left, after applying our model. In order to evaluate the performance of our method, we compute the histograms of the residual errors of the estimated depth values and ground truth range after normalization. The corresponding error histograms are shown in the third column, the x-axis represents the residual errors while the y-axis are the number of pixels with same residual error. For Case I, the highest frequencies are accumulated in the smallest residual errors, thus indicating a good result. For Case II, it is more noticeable the color loss in the objects due to the high degradation coefficient used. This is reflected in the histogram as there exist a considerable amount of pixels whose residual errors are bigger than the previous case. These pixels are located mainly in the background regions of the scenes.

From the results presented several important aspects can be noted. In principle, we can see that in all cases, the estimated depth for images with lower level of degradation is very close to the original value. Second, we can observe that our model does not perform well when dealing with objects with texture in their surface. On the other hand, we see that the positional relationship between objects in the estimate scene, is quite similar to that of the original range image.

We can find some advantages by using our method. First, we do not require the typically stereo vision system to estimate depth. This is particularly difficult in underwater images as corresponding features are difficult to match. Our method only needs to know *a priori* the color appearance of objects outside the water. Then, together with its underwater version we can estimate the relative depth information. While this restriction can be seen as a disadvantage, it is important to mention that it actually depends on the final task to accomplish.

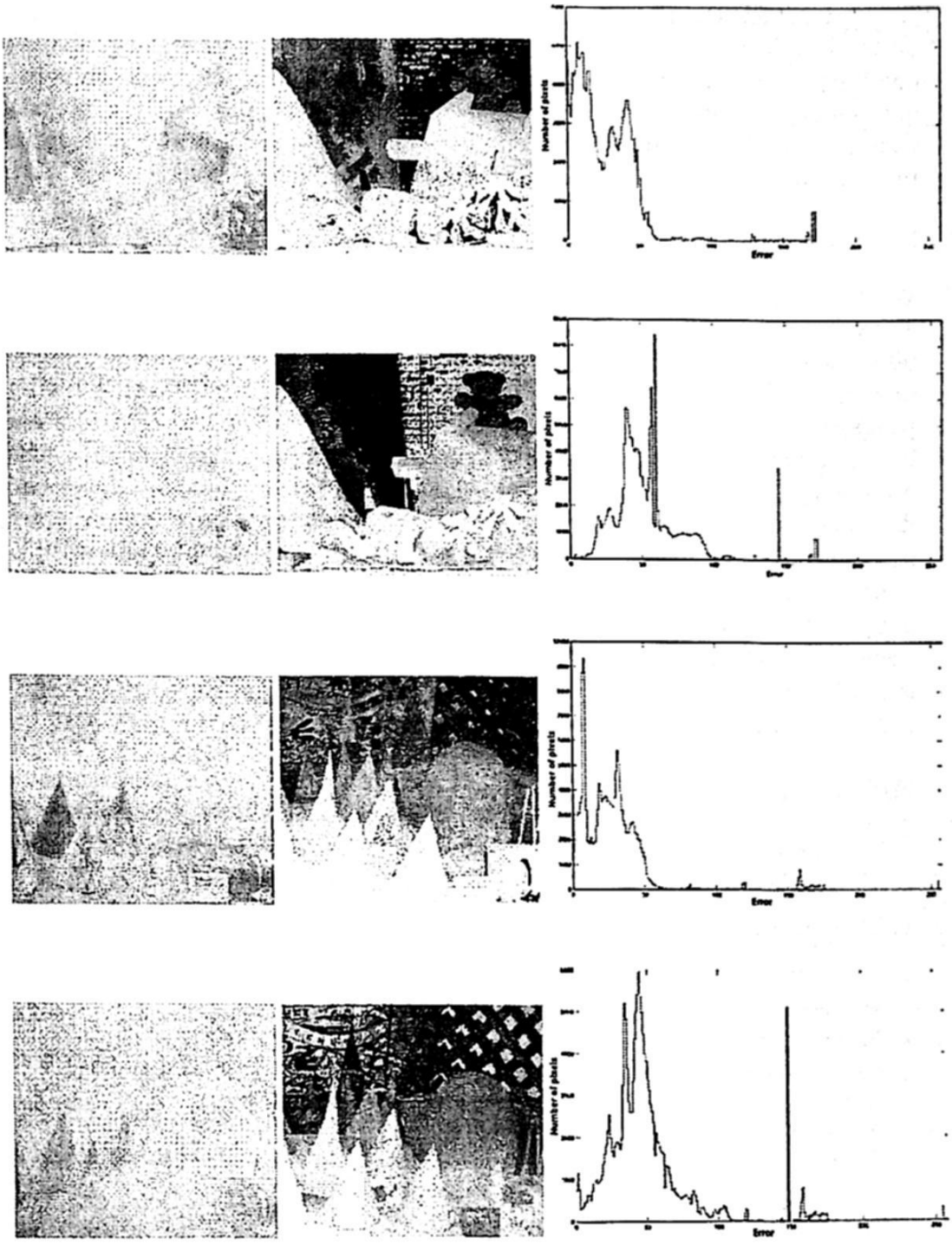


Fig. 3. Experimental results for the depth estimation model. The input aquatic images are in the first column. The resulting depth maps in the second column and in the third, the error histograms when compared to ground truth range maps (shown in Fig. 1.)

In our case, previously known objects (landmarks) will be located in the underwater environment to help a robot to autonomously navigate to a given goal. Another advantage of the model is that having different values for the degradation coefficients at each RGB channel, we can estimate three range maps for one image and eliminate an important amount of the errors present when using only one estimation. These calculations are simple and therefore the execution time is short. This is particular important for real-time applications, especially in underwater environments, where the camera cannot be fixed.

5 Conclusions

We present a method to estimate the relative depth of objects in aquatic images. To evaluate the performance of our method different tests were carried out under different scenarios. On one hand, an image formation model to simulate aquatic images was implemented that reflects the three main factors affecting the light propagation in this type of ecosystem. The photometric characteristics present in the simulated aquatic images, after applying this model, are similar to those observed in images taken underwater. On the other hand, we have developed a depth estimation model based on the image formation model with some simplifications, such as not taking into account the forward scattering present in real aquatic images and considering the color intensity values as a depth cue. In general, we obtain a good estimation of the relative depth of objects in underwater scene, even when testing with images that include this effect. The results are specially good in textureless objects, with few variation in the color intensities. Future work involves using real underwater images to evaluate our model.

Acknowledgements

We would like to thank to Conacyt for funding this research work.

References

1. Boult, T.: DOVE: Dolphin omni-directional video equipment. In Proc.Int. Conf. Robotics and Automation, pp. 214-220 (2000).
2. Foresti, G.L.: Visual inspection of sea bottom structures by an autonomous underwater vehicle. IEEE Trans. Syst. Man and Cyber., Part B, 31:691795 (2001).
3. C. Georgiades *et al.*: AQUA: an aquatic walking robot. Proc. of IROS, Vol. 3, pp. 3525-3531 (2004).
4. Jaffe, J. S.: Computer modeling and the design of optimal underwater imaging systems. IEEE J. Oceanic Engin. 15, pp. 101-111 (1990).
5. McGlamery, B. L.: A computer model for underwater camera system.Proc. SPIE 208, pp. 221-231 (1979).
6. Narasimhan, Srinivasa G., Nayar, Shree K.: Vision and Atmosphere. International Journal of Computer Vision, Vol. 48, No. 3, pp. 233 - 254 (2002).
7. Schechner, Youv Y. and Karpel, Nir: Clear Underwater Vision. Proc. of the Computer Vision and Pattern Recognition, Vol 1, pp. 536 - 543 (2004).
8. Torres-Méndez, L.A., Dudek, G.: Color Correction of Underwater Images for Aquatic Robot Inspection. EMMCVPR, pp. 60-73 (2005).

Fusion of Multiple Mobile Cameras for Object Tracking

Jorge Eduardo Xalteno Altamirano[1], Leopoldo Altamirano Robles[1], Carlos Alberto Reyes[1], Luis Carlos Altamirano Robles[2]

[1] Instituto Nacional de Astrofisica Óptica y Electrónica (INAOE)
[2] Benemerita Universidad Autónoma de Puebla (BUAP)

Abstract:. In this work, we propose a novel way to use sensor fusion based on DSmT approach, in order to track objects using mobile cameras. We use two kinds of camera movements, namely, rectangular and circular movement, and we show that camera movement is a very practical way to deal with partial occlusion. By using accumulative evidence, our approach can detect objects in scenes even if objects are partially occluded, tracking objects where static cameras can not. Experimental results show that our proposal is better than approaches based on static camera sensor fusion for object tracking.

Keywords: Dezert-Smarandache Theory, sensor fusion, mobile sensors.

1 Introduction

Surveillance task can be seen as the process of detecting objects of interest in a sequence of photos (video); these objects can be in motion or they can be statics; if objects are in motion, the task is to follow these objects in order to determine if the behavior or movement of the object is of interest for the observer. Objects tracked are usually people or vehicles; in the case of people, it could be interesting to detect if the person is making a felony, for instance; in the case of automobiles, it would be interesting to determine if the automobile is committing a traffic foul.

Usually, the surveillance of places is done with a sensor placed strategically, in such a way, that the scene of interest is covered properly by the sensor. However, in many scenes it is impossible to cover the whole field of interest with only one sensor. Therefore, it is necessary to use multiple sensors placed properly [7].

In applications that use several sensors, information from one sensor is used to combine and complement the information from other sensor [2]; There are several ways to combine information from multiple sensors as Dempster-Shafer or Dezert-Smarandache approaches [3]; in this work, we use the last one approach because it was proved that is better than other ones, for working with uncertain information [3].

There exists another problem, it's the one of object partial occlusion, it happens when an object is placed between the target and the camera for a while, there are approaches to

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 37-44

solve this problem that use a predictor to approximate the position of the object, however, if the object has a non-predictable movement, the approximation will not be accurate.

In this work we propose an approach based on the cameras movement and in sensor fusion to solve this problem, this approach helps to follow the object and to place it more accurate on the scene.

2 Sensor Position and Movement

The sensors are placed in such way that they cover the same area viewed from different places; in this way, they can have a different sight of the scene and contribute with their information to the sensor fusion. We used two sensors to do our experiments; these sensors were moved in two ways: rectangular and circular styles of movement. These kinds of movement were selected not only for experimental proposes, but also for analyzing different sources of evidence.

In Fig 1.a we show the first kind of movement used in this work (rectangular); in these case the sensors move from side to side horizontally; the movement describes a straight line and the sensors are moved with a regular speed.

The second kind of movement is in circular way; in this kind of movement the sensors move around a point placed in the center of the scene. Fig. 1.b shows a scheme of this movement.

It is important to mention that with mobile sensors, we can cover a wider Field Of View (FOV), in opposition to the approaches where the sensors are static. The movements are based on real life, when a person wants to see an object that is occluded by another object; the person usually moves its head to be able to see the target; in the case of sensors, the approach is the same: we move the sensor to have a better FOV but we also use the movement to deal with the problem of partial occlusion accumulating evidence during movement of sensors for object tracking.

3 Object Movement Detection

The object movement in a sequence of images can be seen as the change of coordinates of an object in the current image respect to the previous image. The movement detection is an important step in many autonomous surveillance systems. In this work, we used two different ways to detect object movement.

3.1 Images difference

The most obvious method to detect changes in a sequence of images is to compare two corresponding pixels to determine if they have the same gray value. In the simplest way, a binary difference $DP_{jk}(x,y)$ between two images $F(x,y,j)$ and $F(x,y,k)$ is obtained by:

$$DP_{jk}(x,y) \begin{cases} 1 & \text{if } |F(x,y,j)-F(x,y,k)|>r \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where r is a threshold to be defined [6]. In the difference of images the pixels with value 1 are considered as the result of the object movement.

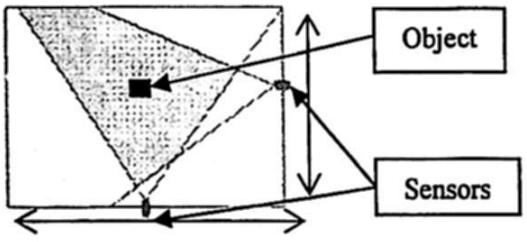


Fig. 1.a) Horizontal sensor movement

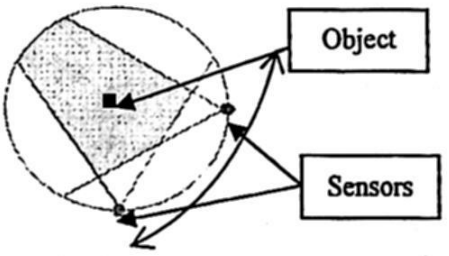


Fig. 1.b) Circular sensor movement

3.2 Size Filter

The result of the previous step is a binary image contaminated with noise; this noise is due to small changes between images; to eliminate this noise, we used a size filter. Simply, the pixels that do not belong to a minimum size group of connected pixels are eliminated.

4 Belief Creation

4.1 Object Characteristics

After the movement detection, we calculate characteristics of the object in the images. Namely: the vertical axis and the first and second statistical moments.

The vertical axis is calculated after the object is detected as [1]; vertical axis is obtained from the outer rectangle (the smallest rectangle that rounds the object), determining the line that crosses the rectangle from up to bottom exactly in the middle. An example of the vertical axis is shown in the (Fig. 2).

Fig 2. Outer rectangle and vertical axis of an object.

Other side, the statistical moments are calculated as follows:

$$m_{pq} = \sum_{x}\sum_{y} x^p y^q f(x,y) \tag{2}$$

The first and second statistical moments are m_{00} and m_{01} respectively.

4.2 Believes or Masses

A belief is a data that tells us the certainty of an object to exist in the image [3]; it can be derived from any object characteristic. In this work, it has been done using the characteristics mentioned above.

For the vertical axis, the whole image is partitioned in cells with 20x20 pixels in size. Then believes are created according to the cells that the vertical axis crosses. It means that if the vertical axis passes through three cells, then there are three believes. The number of cells where the axis passes is the number of elements of the frame of discernment; this frame will be used in the fusion step. The values of the elements in the frame are the number of pixels that belong to the vertical axis in that cell.

For the case of moments, as we consider only two momentums, therefore there are only two elements in the frame of discernment. The values of these elements are the values of the moments.

Both cases, the vertical axis and the moments, are normalized so that the quantities of their elements are equal to 1.

5 Sensor Fusion

5.1 Dezert-Smarandache Theory (DSmT)

In 2002 Jean Dezert and Florentin Smarandache proposed a novel Information Fusion Model [3]; it is an evolution of the Dempster-Shafer model. The DSmT uses a frame of discernment Θ; this frame is a set of propositions $\Theta=\{\theta_1, \theta_2, ..., \theta_n\}$, that they are used to carry out the information fusion, DSmT has three main characteristics:

- The evidence combination refutes the third logical principle; this principle establishes that something "is or is not". The DSmT does not require this principle, therefore it can consider concepts related to the fuzzy logic.
- It is proposed a new combination rule.

- The elements from the frame of discernment are not necessarily exhaustive and exclusive.

The model DSm considers that the frame of discernment is dynamic. In this case some elements from Θ could be exclusive at a time, and they can not exist anymore at other time; this model is known as the free model ($M^f(\Theta)$) of the Dezert-Smarandache Model.

5.1.1 Classical DSm Combination Rule

When the free model $M^f(\Theta)$ is used, the combination rule used for a frame Θ with k elements is defined as in the equation (3):

$$m_{M^f(\Theta)}(C) = \sum_{\substack{x_1,x_2,\ldots,x_k \in D^\theta \\ x_1 \cap x_2 \cap \ldots \cap x_k = A}} \prod_{i=1}^{k} m_i(X_i) \tag{3}$$

6 Proposed Solution

The proposed solution has been implemented in a hardware/software prototype, it has served to show the effectiveness of the approach in real situations, even when in some experiments the scene is in a small scale; this does not gives less generality to the system, minimum adjustments have to be done for cover any other scene

The scene included objects in movement (people, animals, cars, etc), as well as static objects (trees, buildings, parked cars, etc), the system has operated in outdoor conditions. In order to carry out camera movement, we used one template for each position of the cameras.

6.1 Horizontal Movement

For the case of the horizontal movement of the camera, we use a sequence from PETS[5]; this sequence was taken with two cameras. The movement of the cameras is 5 pixels each frame. The cameras move from left to right and vice versa. For each camera we took some frames as templates in order to obtain the binary image. Once we have binary image, we used a size filter to clean the image. The result of the filter is, in almost all the cases, the tracked object. Fig. 4 shows the i-th template and Fig. 5 shows the i-th frame; the Fig. 6 shows the difference between the template and the frame (binary image).

In the binary image, we found the vertical axis and built the frame of discernment as explained in section 4.

In order to carry out the sensor fusion, the two sensors used in the system have to give information; if one of the sensors does not, the information from the other sensor is used as the output due to the inexistence of help from the other sensor.

Fig. 4. I-th template.

Fig. 5. I-th frame.

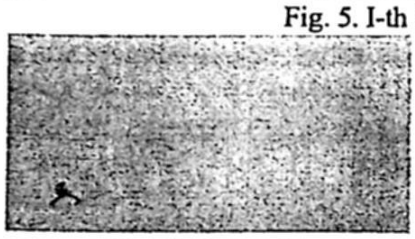


Fig. 6. Binary image.

The sensor fusion is done with the two frames and the output is an array of values; we take the largest value as the final output and the output tell us where the object is placed. As we know where the object is, we can evaluate the final result.

6.1 Circular Movement

In order to prove that the system works in other kind of scenes, we made a device that allows us to make a circular movement (Fig. 1.b). In this device, we placed two cameras placed 1.5m from the center of the scene and they have 90° of separation between them. In this experiment, we notice that the movement could serve to solve the problem of partial occlusion of the object.

6.2 Sensor Movement as a Solution of Object Occlusion

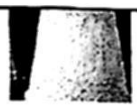
Such as a people moves the head in order to see an object occluded, the cameras movement of the sensors can be used to see an object that is partially occluded by another object. Notice that as the sensor moves, it sees a different FOV; the movement helps the sensor to see other interesting things; it is shown in table 1.

6.3 The Sensor Movement Increases the Evidence

The main difference between the work in [2] and our work is that the movement of the sensors contributes to obtain a larger evidence of the object; as the sensor moves, we make information fusion in each sensor; after 3 frames, the information that has been

fusioned and accumulated in each sensor (accumulated evidence) is fusioned. The results show that the fusion of the accumulated evidence is better than just the only fusion in each frame.

Table 1. The movement of the cameras can be used to discover occluded objects.

Camera 1	Camera 2
	
	
	
	

6.4 Experimental Results

In the Table 2 we show that evidence grows with the accumulated fusion; it is due to the movement of the sensors: when they move, they can see a lager part of the object. For the time t and $t+1$ in Table 2 there are no information in the accumulated evidence, this is because the fusion between the sensors is not performed in these times; however in the time $t+2$ the evidence is larger than the single fusion in t, $t+1$ and $t+2$.

Table 2. Results of the accumulated sensor fusion versus the single sensor fusion.

	t	$t+1$	$t+2$
Single Fusion	[0.9106 0.0894]	[0.9819 0.0181]	[0.9106 0.0894]
Accumulated evidence	[0.000 0.000] (no sensor fusion)	[0.000 0.000] (no sensor fusion)	**[0.9982 0.0018] (Fusion)**

In table 3 the required time to calculate sensor fusion is shown. We can see that the accumulated fusion takes more time than the simple fusion; however the results are better with the accumulated fusion, respect to simple fusion.

We have compared the fusion based on DSmT versus an approach based on Bayesian fusion; the results are that the DSmT is more accurate than the Bayesian fusion due to the DSmT uses more information than the Bayesian approach. Figure 7(a) shows the regions were the object is and Figure 7(b) shows the results between the DSmT (red) and the Bayesian fusion (blue)

7 Conclusions

We have shown a novel approach to make mobile sensor fusion. The obtained results show that our proposal improves the evidence obtained respect to static sensors, due to the accumulation of evidence in terms of previous scenes. This is of especial interest when the objects are partially occluded respect to a fixed position of the sensors and therefore, the evidence generated by the fusion is almost null or totally null. Allowing the sensors to move, the system can discover partially or completely occluded objects, though they were static respect to the plane sensed.

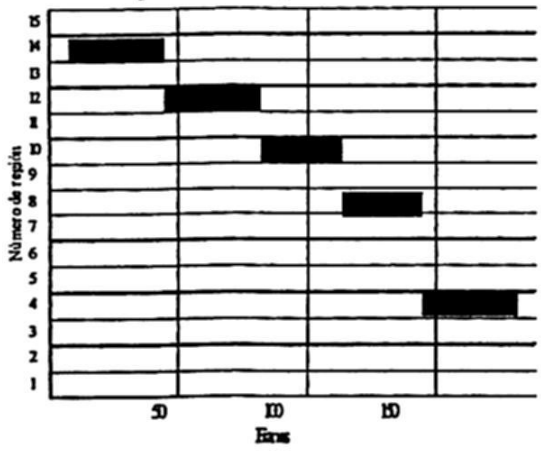


Figure 7 (a) Real object positions

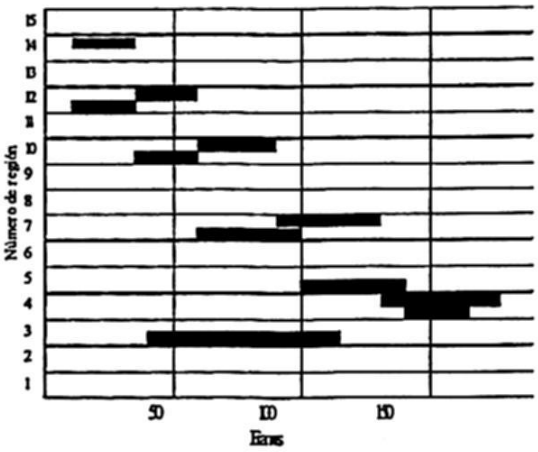


Figure 7 (b) Object position reported by DSmt (red) Bayesian fusion (blue)

References

1. Weiming Hu, Min Hu, et. Al: "Principal Axis-Based Correspondence between Multiple Cameras for People Tracking". IEEE Transactions on pattern analysis and machine inteligence April 2006
2. Esteban García (2008), "Fusión multicámara para seguimiento de objetos basada en Teoría evidencial". Tesis de maestría INAOE.
3. Floretin Smarandache, Jean Dezert: "An introduction to the DSm Theory for the combination of paradoxical, uncertain and imprecise sources of information". Online paper revised on August 1st 2006,
4. Aridiam Biswas, Pritwijit Guha et al: "Intrusion Detection and Tracking with Pan-Tilt Cameras". Second European conference on Computer Vision 1995.
5. PETS Repository: an Internet resource. Available from anonymous ftp: pets2001.cs.rdg.ac.uk/. University of Edinburgh..
6. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, "Digital Image Processing using Matlab", Pearson Education.
7. Boudet, L., Midenet, "Pedestrian crossing detection based on evidential fusion of video-sensors", 2009 Transportation Research Part C: Emerging Technologies, Article in Press

A Bayesian Approximation of a Computational Model of the Visual Cortex

Elias Ruiz, L. Enrique Sucar

Instituto Nacional de Astrofísica Óptica y Electrónica
Luis Enrique Erro 1, Tonantzintla, Pue., Mexico
{elias_ruiz, esucar}@inaoep.mx

Abstract. Recently computational models of the visual cortex have become more detailed, however, they do not include important aspects of visual systems, in particular their property of not only propagating information bottom-up, but also top-down. They do not provide a clear theoretical understanding to analyze the advantages and limitations of these models. We consider that the Bayesian framework will help in both aspects, and propose an initial Bayesian approximation to a computational model of the visual cortex. The Bayesian model is hierarchical; each stage is implemented as a set of Bayesian classifiers that emulate the different types of cells in the original model. Experiments in object detection with both models show similar results in terms of accuracy; however the Bayesian model is about twice as fast. This Bayesian implementation opens the door to a better understanding of these models, and to a future representation as an integrated graphical model.

1 Introduction

Since the pioneering work of Hubel and Wiesel [?], there have been important advances in the understanding and computational modeling of the visual cortex. In particular, the model proposed by Serre, Poggio and others [?,?] has shown impressive performance in several visual tasks such as object identification. However, biologically inspired models, in general, do not include important aspects of natural visual systems, in particular their property of not only propagating information bottom-up (from the image), but also top-down (to incorporate previous knowledge of the domain). Also, they do not provide a clear theoretical understanding which will allow us to analyze in more depth the advantages and limitations of these models.

In this paper we develop a Bayesian representation of the computational model of the visual cortex based on [?][1]. We consider that the Bayesian framework will help in both aspects, a better theoretical understanding and a more flexible implementation which can incorporate *a priori* knowledge.

The proposed model is based on a hierarchical representation, analogous to the model of [?], composed of alternating layers of simple and complex cells.

[1] We are not proposing a new biological model, but a Bayesian approximation of the model of Serre, Poggio and others [?].

© J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad and H. Sossa (Eds.)
Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 45-52

In the Bayesian approximation, each cell is implemented as a naïve Bayesian classifier (NBC) which emulates the different types of cells. The combination of these layers of Bayesian classifiers makes the complete model. Learning consists of estimating the parameters of certain layers (the others are fixed by design) based on positive and negative examples, and inference is basically done by probability propagation at each layer. Although in the current implementation inference is done separately at each layer, it is possible to integrate all the layers in a single Bayesian network.

We compared both models in terms of accuracy and efficiency in object detection, for different classes of objects in the Caltech-256 database [?]. The experiments show similar results in terms of accuracy, but the Bayesian implementation is more than two times faster than the original one. We also evaluated the robustness of both models in terms of rotation and scale, showing better invariance to scale and less to rotation, which is consistent with the model of [?].

2 Models of the visual cortex

Since the work of Hubel and Wiesel [?], there have been important advances in the understanding of the visual cortex [?,?,?]. Our work is inspired on model of Serre and Poggio [?]. This model proposes layered structure similar to the arguments presented by Hubel and Wiesel. One objective of this model is to recognize objects in images based on a prior training analogous to the learning process in primates. This model recognizes patches (image regions) which are reduced increasingly in each layer until it reaches an object categorization stage from several images. One category of objects is a set of objects that share similar characteristics, such as cars. In this way, this model recognizes objects using a model inspired on the visual cortex of primates. The main objective is to obtain a better understanding of the primate's (and human) visual system.

A computer implementation of the model of [?] has been developed. It is a hierarchical model composed of alternating layers of simple and complex cells. Each layer resembles certain types of neurons of the biological visual system. Its implementation has two phases, training and testing. In its simplified version, three layers are applied in the training phase: S1, C1 and S2. The S1 layer is the application of a Gabor filters [?] in different directions and scales. The C1 layer receives the results of the previous layer and performs a set of Max filters. Finally, in the S2 layer, patches are extracted from the sets of images generated in the two previous layers. In test phase, the same layers S1 and C1 are applied (Gabor filters and Max filters). In layer S2 the patches are evaluated to obtain a similarity measure. The results are passed to Max filters and finally to a classifier (Figure ??). Next we present a Bayesian approximation based on this model.

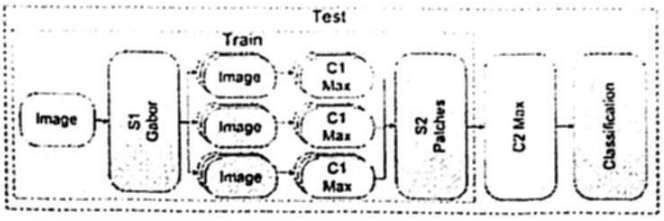


Fig. 1. Block diagram of train/test phases of [?]. In S1 a set of Gabor filters, and in C1 Max filters are applied, in the same way for both phases. In S2 the patches are extracted for the train phase and evaluated for test images. A set of Max filters is applied in C2 for the test phase only. Finally, in the classification layer the images are divided into positive and negative using a nearest neighbor classifier.

3 Bayesian Model

3.1 Model Layers

The probabilistic approximation is based on the simplified model proposed by Serre and Poggio [?]. This model has layers S1, C1, S2, C2 and a classification layer. The probabilistic model proposed uses naïve Bayesian classifiers (NBCs) in order to emulate the different types of cells in each layer, and sustain the properties of the computational model.

The model presents a hierarchical structure with the possibility of forming a larger network. In the present approach, each layer is implemented as a set of NBC's. A general outline of the different layers, from the image to classification is depicted in figure ??. The main layers are the following: S1 (Gabor filters), C1 (Max filters), S2 (Patches), C2 (Max filter) and Classification. In the next subsections, each layer is described.

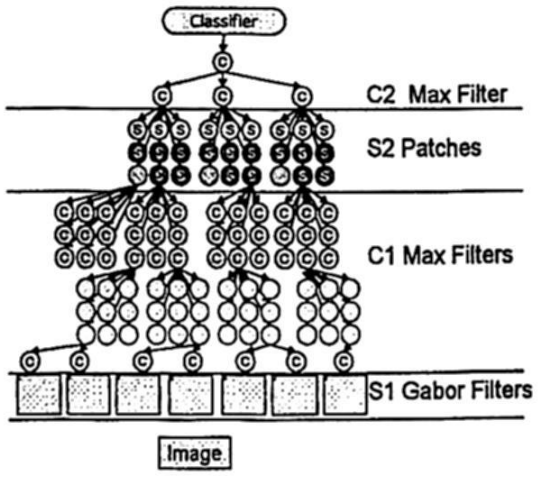


Fig. 2. General outline of the proposed probabilistic model. It has 5 layers. Layer C1 has 3 phases. The hierarchical structure of the model is composed of layers of NBCs which can be integrated in a global network.

S1 layer This layer contains a bank of Gabor filters as on the model of Serre and Poggio [?]. Neurons that correspond to S1 are sensitive to different edges of an object in an image. A Gabor filter is a type of Gaussian filter which extracts edge features at different orientations and scales [?]. A series of images generated by a pair of scales are called a band, for all the orientations considered.

C1 Layer The C1 Layer performs three operations with each band generated by the S1 layer. First, a Max filter is applied to each pair of images of a band with the same orientation (but slightly different scale) to obtain the maximum of the pair. In the second phase, a series of local maximums are calculated using a sliding window, the number of locals maximums is the same that the image size selected in the previous step. In the third phase of the C1 layer, a sub-sampling that eliminates some maximums pixels is performed. In the proposed model, the Max filter is implemented as a NBC, which provides flexibility to change the parameters or extract information from different pixels. A NBC is defined by a set of attributes and a class, assuming that the attributes are conditionally independent given the class, this model considers the class and attributes as discrete variables. A graphical representation of a NBC is illustrated in Figure ??-a. Each attribute has an associated table of conditional probability given the class, and there is also an *a priori* class probability. An example of a conditional probability table (CPT) is presented in Table ??. Thus, the Max filter is modeled with a NBC, where the class defines the probability of finding a maximum among a set of pixels. The attributes are pixels values, and for the local Max, in a certain neighborhood (e.g., 3x3 pixels region). Probability calculation is given by Bayes rule:

$$P(Max|C_1, ...C_n) \approx P(Max)P(C_1|Max)\dots P(C_n|Max), \quad (1)$$

where $C_1, \dots C_n$ are pixels in the edge image and Max is class probability. Each value $P(C_k|Max)$ is given in proportion to the pixel intensity (for images in grayscale). Dark pixels will have a low probability, while bright pixels will have a high probability. Given that the class is binary, the counterpart of Max is named Min. It is proposed an increasing exponential probability function for class Max and a decreasing exponential probability function for class Min. The sub-sampling is performed using also a NBC, illustrated in Figure ??-b. Pixels that do not provide meaningful information have a uniform probability distribution. An example of the CPT for node A_1 is presented in Table ??.

S2 Layer The S2 layer calculates a metric to estimate the similarity of an image region of the training set, processed by C1 (called patch), with another region of a test image considering a distance value. In the Bayesian model, the patch similarity metric is based on the assignment of higher probability values when the pixels in the patch and test region pixels are consistent. Given that a pixel has a discrete and finite number of states, the probability for each state is given by a normal distribution:

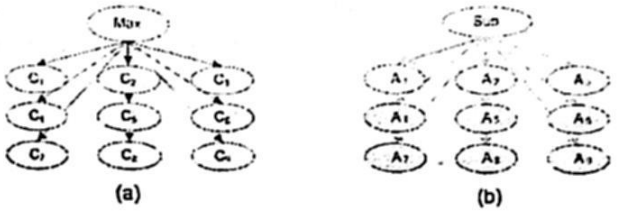


Fig. 3. a) NBC for the Max local phase. Each node has the same CPT. b) An example of the NBC for the sub-sampling phase. Only the A_1 node provides meaningful information.

Table 1. Up Table: CPT for each attribute (C_i) in Max NBC. Down Table: CPT for the A_1 node of the sub sampling filter (figure ??-b). Both consider only 8 values as pixels are discretized into 8 states.

Gray scale	1	2	3	4	5	6	7	8
Max	0.094	0.096	0.103	0.113	0.122	0.132	0.150	0.186
Min	0.185	0.181	0.166	0.148	0.129	0.111	0.074	0.003

Sub_+	.002	.034	.068	.102	.136	.170	.217	.269
Sub_-	.228	.202	.173	.144	.115	.086	.046	.002

$$x_n = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x_n-\mu)^2}{2\sigma^2}}, \tag{2}$$

where x_n represents a probability value that corresponds to each of the n states that a pixel can have (8 gray levels), μ is the value of the state that receives the highest probability (corresponds to the training patch pixel value) and σ is the standard deviation. The formula should be applied to each pixel in each patch extracted. The σ value is obtained empirically; with a value of $\sigma < 0.3$ we obtained a good performance. As the operation is carried out for every pixel of the selected patch, the NBC has as many nodes as pixels in a patch. The evaluation of this NBC with images regions generates a set of probabilities proportional to the similarity values of the patch given in the original model.

C2 layer The C2 layer is composed of Max filters implemented also as NBCs in a similar way as the C1 layer. The goal is to recognize the highest probabilities that correspond to the most similar patches. The outputs from layer S2 are discretized before entering the C2 layer.

Classification The final stage is a classifier that determines the probability of the presence (+) or absence (-) of certain object in the image, and it is given by:

$$P(+|R) \propto P(+)P(R_1|+)\ldots P(R_n|+) \tag{3}$$

$$P(-|R) \propto P(-)P(R_1|-)\ldots P(R_n|-), \tag{4}$$

where R_k represent a specific patch. $P(R_k|+)$ and $P(R_k|-)$ are the probabilities given by a patch of the presence or absence of a class of object. The threshold for categorizing an image (presence or absence of an object) after normalization is set to 0.5.

4 Experiments and results

Fig. 4. Examples of categories of images from the Caltech-256 collection.

In the experiments we compared empirically the original model and the Bayesian approximation, and also evaluated the tolerance of both models to different orientations and scales. For this we used the Caltech-256 collection [?], which contains images for 256 object classes, of which we selected motorcycles, faces, cars, animals, among others. Examples of some of the test images are presented in Figure ??. We used the MatLab implementation of the simplified model provided by [?], and the Bayesian model was also implemented in MatLab.

4.1 Results

The comparison of the Bayesian model with the original one is performed using equivalent parameters for both models, and the results are summarized in Table ??. The table shows the average recognition accuracy for different classes of objects. The numbers in parentheses are the percentage of hits on the positive and negative examples in the test sets. Approximately 100 to 400 images were used in each test (50-200 for training and 50-200 for testing).

We tested with patches of different sizes, from 5x5 up to 16x16 pixels. Those that obtain better results are patches of 10x10 pixels. The number of patches used was varied from 10 patches up to 250. Smaller patches increase false positives and the use of a few patches presented poor results. The results presented in the tables consider 100 patches. The classification accuracy is slightly improved by the use of more than 100 patches.

From the table we observe that the accuracy of both models is similar for all types of objects, with no significant difference in average. Both models have a slightly higher accuracy for the positive class.

Rotation Invariance We evaluated the tolerance of both models to changes in orientation (see table ??). Both models still have a good precision when the change in orientation is less than 15°. With higher changes, classification accuracy decreases in a significant way.

Table 2. Recognition rates for the original and Bayesian models for different types of objects in the Caltech-256 collection. In parentheses we show the classification accuracy for the positive and negative subsets, and time for train and test phases.

	Bayesian model	Original model	Bayesian time	Original time
motorcycle	90% (100+,81-)	89% (94+, 85-)	17 min	36 min
tire	72% (84+, 60-)	71% (77+, 64-)	8 min	17 min
faces	85% (100+, 71-)	83% (100+, 66-)	16 min	34 min
cars	81% (93+, 69-)	85% (100+, 70-)	9 min	18 min
zebra	85% (100+, 71-)	83% (91+, 76-)	11 min	23 min
similar faces	92% (100+, 84-)	99% (100+, 98-)	18 min	37 min
average	84% (96+, 72-)	85% (92+, 76-)		

Scale Invariance In a similar manner, we also evaluated the robustness of both models to changes in scale (see table ??). We find out that the results are better if the scale is increased and both models are less robust to a reduction in scale. Sizes smaller than 50% of the original images produce poor results. As one would expect, changes in scale only affect the positive examples.

4.2 Analysis

From these experiments we observe that both models produce similar results in terms of accuracy; comparable to other methods in the state of the art. The quantization of the image (discretization) in the Bayesian implementation does not affect the accuracy of the model. Also, both models have a higher tolerance to changes in scale than changes in orientation, which is consistent with the model of visual cortex. In terms of computation time, the Bayesian model registered in average 2.5 seconds per image, and the original model 5 seconds.

Table 3. Classification accuracy for the motorcycle images with different orientation.

	Bayesian model	Original model
motorcycle 0°	90% (100+, 81-)	89% (94+, 85-)
motorcycle 5°	88% (96+, 81-)	89% (93+, 85-)
motorcycle 15°	84% (87+, 81-)	83% (81+, 85-)
motorcycle 90°	59% (37+, 81-)	58% (30+, 85-)

5 Conclusions

In this paper we develop a Bayesian representation of a computational model of the visual cortex, based on a hierarchical representation composed of layers of different cells; each cell is implemented as a NBC. Learning consists of estimating the parameters of the NBCs for certain layers, and inference is done by

Table 4. Classification accuracy for the face images with different scales.

	Bayesian model	Original model
similar faces	93% (100+, 86-)	99% (100+, 98-)
faces 120%	92% (98+, 86-)	97% (96+, 98-)
faces 150%	90% (94+, 86-)	94% (89+, 98-)
faces 80%	85% (84+, 86-)	92% (86+, 98-)
faces 50%	63% (40+, 86-)	67% (36+, 98-)

probability propagation at each layer. Although in the current implementation inference is done separately at each layer, it is possible to integrate all the layers in a single Bayesian network.

We compared our model and the original one in terms of accuracy and efficiency in object detection. The experiments show similar results in terms of accuracy; however the Bayesian implementation is two times faster. We also evaluated the robustness of both models in terms of rotation and scale invariance, showing better invariance to scale.

Our main contribution is a Bayesian implementation of a model [?] of the visual cortex that opens the door to an understanding of these models under a Bayesian framework. For instance, we observe that certain types of cells are approximated by exponential distributions and others by normal distributions. Of course more work is required at higher levels of description to obtain a complete Bayesian interpretation of the bio-inspired models of the visual system. In the future we expect to implement this model as a global Bayesian network which will allow including prior knowledge via top-down propagation.

References

1. Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.
2. D. Gabor. Theory of communication. *JIEE*, 93(3):429–459, 1946.
3. G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.
4. D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *J Physiol*, 195(1):215–243, March 1968.
5. T. Serre, L. Wolf, S. M. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(3):411–426, March 2007.
6. T. Poggio T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman. A theory of object recognition: computations and circuits in the feedforward path of the ventral stream in primate visual cortex. Technical Report CBCL-259, MIT Artificial Intelligence Laboratory, December 19 2005.
7. Doris Y. Tsao, Winrich A. Greiwald, Tamara A. Knutsen, Joseph B. Mandeville, and Roger B. H. Tootell. Faces and objects in macaque cerebral cortex. *Nature Neuroscience*, 6(9):989–995, September 2003.

Refined Geometric Camera Calibration using Linear Simplex Method

Enrique Rivera[1] , Rafael Lemuz[1], Carlos Guillén[1] , Irene Ayaquica[1] and Salvador Alcantara[2]

[1] Facultad de Ciencias de la Computación, BUAP,
[2] Centro de Investigaciones en Dispositivos Semiconductores, BUAP,

[1] {ecastilo,rlemuz,cguillen,ayaquica}@cs.buap.mx, [2]salvador@siu.buap.mx

Abstract. A camera is considered calibrated if the focal distance, principal point and lens distortion parameters are known. The level of accuracy of the measurement of these parameters has always been a fundamental factor in stereo metrology. In this work we present a calibration algorithm for microscopic images where optical aberrations degrade the accuracy of traditional methods. A linear refinement steep is added to a state of art algorithm to improve the accuracy of the current methods. Experimental results show the quantitative improvement of the proposed method in a volume of inspection of 60x60x126 μm using both, synthetic and real images.

Keywords: camera calibration, simplex method, linear optimization.

1 Introduction

Geometric camera calibration is a necessary prerequisite for recovering the 3D position of a scene point when only its projection in two images captured from different viewpoint location is known, in this case, each projection defines a ray in space and the intersection of both rays is precisely the 3D point location.

The key idea behind calibration is to write the projection equations linking the known coordinates of a set of 3D points and their image projections, and solve for the camera parameters. This mean the determination of the projection matrix P which encompass both, intrinsic and external pose parameters (camera's position and rotation in the world coordinate frame) modeling the image formation process. Usually, radial and pincushion distortions are also modeled using additional parameters. In this stage, the estimated parameters relate the 3D coordinates of a point of the scene, and the 2D coordinates of the projected point of the image. However, when microscopic images are captured, lens optical aberrations introduce additional image distortions which must be considered.

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 53-58

Numerous methods for camera calibration have been developed, each one having its pros and cons. Pioneer methods use 11 parameters to define the camera projection matrix P and optimize a cost criterion to compute those parameters. Linear methods use linear algebra tools to find and initial solution [1,2], Non-linear methods refine the solutions to improve the estimate [3]. A recent linear method [4] based on the Gintic camera model originally proposed in [2] models the image formation process using 16 parameters has shown its superiority when considering microscopic images.

2 The Gintic Camera Model

The mapping between 3D space coordinates $W = (x_n, y_n, z_n)$, and 2D image locations $w = (p_x, q_x, s)$ in homogeneous coordinates are expressed using the Haralick Reduced model proposed in [2] by

$$\begin{matrix} p_x \\ q_y \\ s \end{matrix} = R(\omega, \phi, \kappa) \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} \quad (1)$$

where ω, ϕ and κ are the rotation angles for the x, y and z axes respectively, then after some algebraic substitutions by using a perspective camera model with a_u and a_v the focal length in x and y axes respectively; k_1 and k_2 representing radial geometric distortion of lens, and (u_p, v_p), principal point in the image coordinates the following expression can be derived:

$$(1 + k_1 r_n^2 + k_2 r_n^4 + k_3 r_n^6) \begin{pmatrix} u_n - u_p \\ v_n - v_p \end{pmatrix} = \frac{1}{s_n} \begin{pmatrix} a_u p_n \\ a_v q_n \end{pmatrix}, \quad (2)$$

where r_n^2 is the distance of the pixel at (u_n, v_n) to the principal point in the image plane given by:

$$r_n^2 = (u_u - u_p)^2 + (v_u - v_p)^2 \quad (3)$$

Finally the two linear equations of the Gintic calibration model are:

$$\begin{aligned} a_1 x_n + b_1 y_n + c_1 z_n + d_1 u_n + e_1 u_n x + f_1 u_n y_n + g_1 u_n z_n + 1 = 0 \\ a_2 x_n + b_2 y_n + c_2 z_n + d_2 v_n + e_2 v_n x + f_2 v_n y_n + g_2 v_n z_n + 1 = 0 \end{aligned} \quad (4)$$

Since each point imaged from the calibration pattern contributes in two linear equations, by stacking 2n linear equations a linear system can be constructed and solved to bring an estimate of the camera projection parameters involved in the image formation process.

Thus, from (3) and (4) it can be observed that 16 parameters are considered to model the projective mapping of a single camera. Then, using linear algebra tools a solution can be estimated when an over determined set of equations is known, see [1] for details. However, due to image noise and image quantization, the initial estimate is prone to limited accuracy we propose to add a second stage to improve the solution using the Simplex method.

3 Gintic Camera Calibration Refinement

Given the initial linear solution we compute a refinement over the Gintic camera parameters.

An error function is defined based on the measurement of the geometric distance in the image plane.

3.1 Proposed Optimization Error Function

The error function called the reprojection error, considers the difference between the measured image coordinates w_i and the estimated image coordinates $\widehat{w_i} = PW_i$ computed using the W_i 3D coordinates and the parameters of the Gintic camera model P.

We use the notation $d = \|w_i - \widehat{w_i}\|$ to denote the Euclidean distance between the points represented by w_i and $\widehat{w_i}$. Then, the total error for the set of points used in the calibration process is:

$$\sum_{i=1}^{n} \|w_i - \widehat{w_i}\| \qquad (6)$$

We use the simplex method introduced in [5] to minimize the error function (6). The simplex method has the advantage that neither requires the derivative of the error function nor the orthogonal condition needed by non linear optimization methods like least squares [3].

The algorithm starts with an initial basic feasible solution (bfs) and tests its optimality. If some optimality condition is verified, then the algorithm terminates. Otherwise, the algorithm identifies an adjacent bfs, with a better objective value. The optimality of this new solution is tested again, and the entire scheme is repeated, until an optimal bfs is found. Since every time a new bfs is identified the objective value is improved, and the set of bfs's is finite, it follows that the algorithm will terminate in a finite number of iterations.

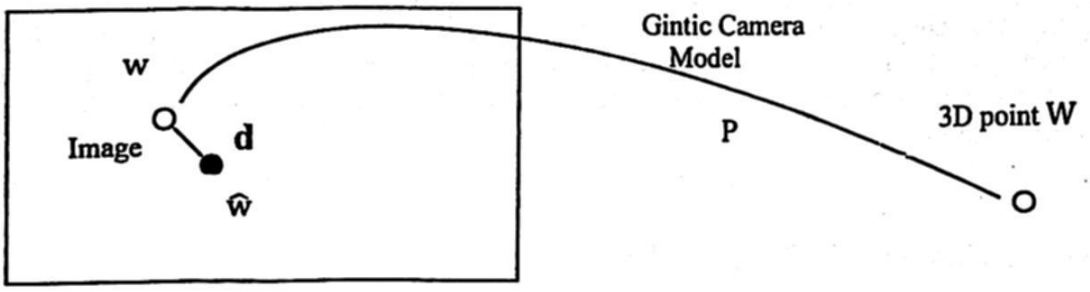


Fig. 1. The graphic representation of the reprojection error.

4. Experiments

Synthetic and real images were used to evaluate the accuracy of the calibration method proposed in this work (see figure 2). In the real setup we have captured four images of a check board calibration pattern using a stereo imaging setup in a volume of inspection of 60x60x120 μm. Then, using only the closest and farther images, the imaging system is calibrated. Ten additional images taken inside the volume of inspection where used to evaluate the real accuracy of the proposed algorithm, detecting and matching corner points and measuring the estimated 3D coordinates by triangulation [4].

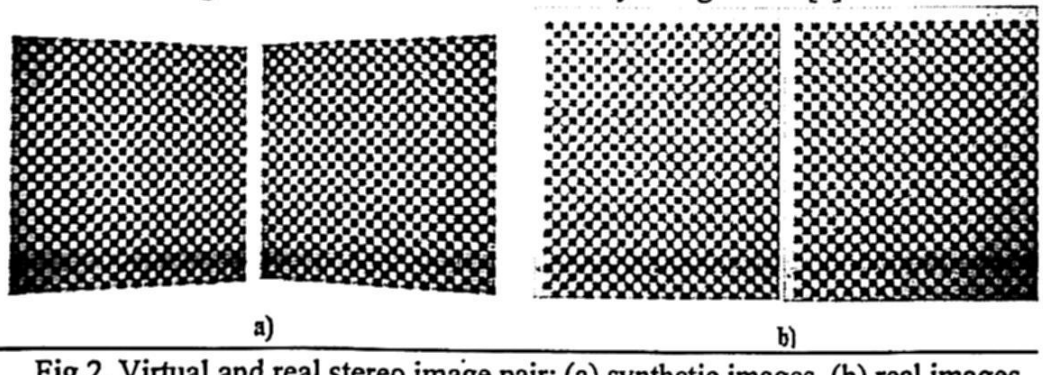


Fig.2. Virtual and real stereo image pair: (a) synthetic images, (b) real images.

The accuracy of the improved Gintic camera model is illustrated by figure 3. In 3-(a) the accuracy of calibration method using synthetic images. Figure 3-(b) shows the accuracy of the proposed method using real images. Note that the method allows accurate measurements. Table 1 summarize the absolute difference in the calibration accuracy using synthetic and real images (The standard deviation absolute difference is about 0.25μm, 0.37 μm, and 0.6 μm for x, y and z axis respectively).

Table1. The absolute difference of statistical variation for synthetic and real data

Comparative Results (All units in μm)						
	X Mean	**Y mean**	**Z mean**	**X std**	**Y std**	**Z std**
Synthetic	-1.87e-12	5.77e-14	3.44e-13	0.064	0.056	0.096
Microscope	-0.02	-0.008	0.142	0.31	0.425	0.711
Absolute Difference	**0.02**	**0.008**	**0.142**	**0.24**	**0.369**	**0.614**

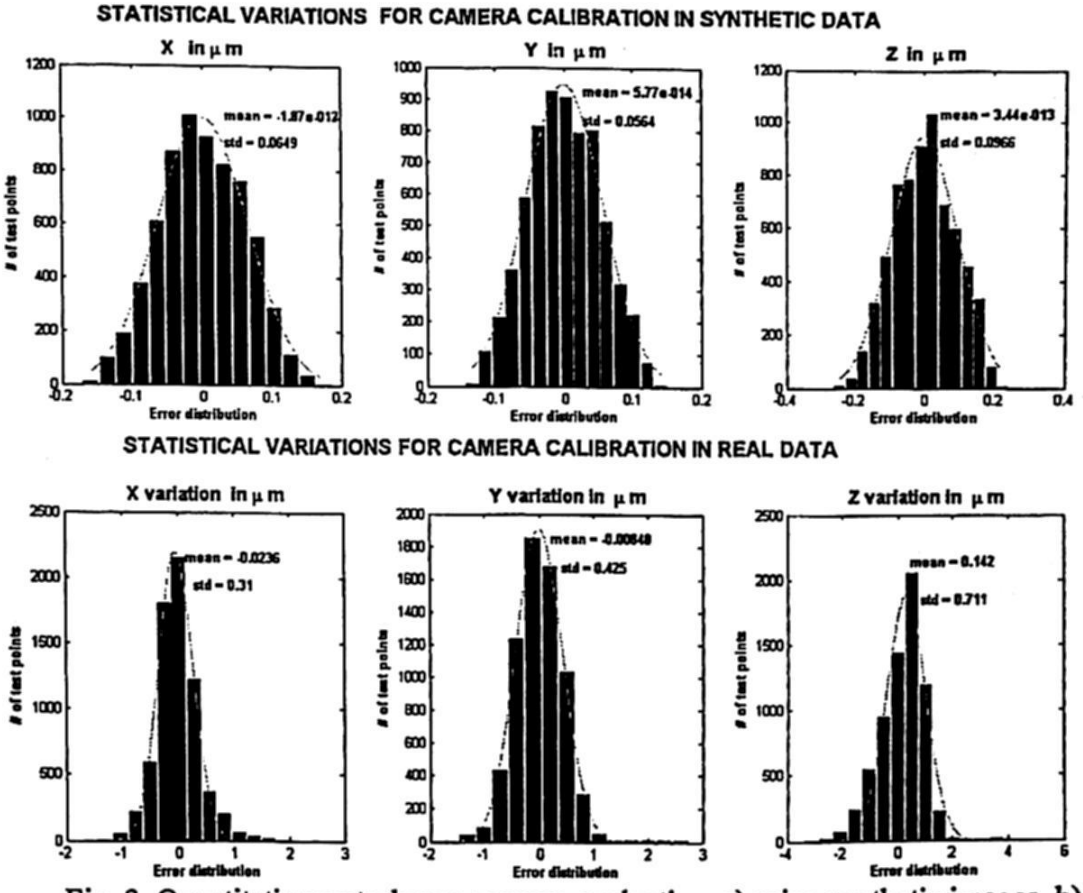


Fig. 3. Quantitative metrology accuracy evaluation. a) using synthetic images, b) evaluation using real images.

5. Conclusions

An improved calibration algorithm was presented; an initial estimate of the camera parameters that model the image formation process using a linear method was first presented. Then, a linear refinement using the Simplex method was derived to improve the estimation of the Gintic camera parameters. The new method was evaluated using synthetic and real images where an improvement of an order of magnitude was obtained. In future work we plan to integrate a preprocessing stage to overcome some of the limitations of the algorithm to take into account higher order optical aberrations in microscopic images.

References

1. Abdel-Aziz, Y.I., & Karara, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry," Proceedings of the Symposium on Close-Range Photogrammetry(pp. 1-18). Falls Church, VA: American Society of Photogrammetry. (1971)
2. Xu Jian, Adrew A. Malcolm, and Fang Zhong Ping, "Camera Calibration with Micron Level Accuracy," Technical Report. (2001)
3. Xu Jian, Adrew A. Malcolm, and Fang Zhong Ping, "3D Digital Photogrammetry," Technical Report. (2000)
4. Beardsley, P. A., Murray, D. W., and Zisserman, "Camera Calibration Using Multiple Images," In Proceedings of the Second European Conference on Computer Vision, Lecture Notes In Computer Science, vol. 588. Springer-Verlag. (1992)
5. Emanuele Trucco, Alessandro Verri, "Introductory Techniques for 3-D Computer Vision," . Prentice Hall, (1998)
6. Nelder, J. A. and Mead, R., "A simplex method for function minimization",(pp, 308-313), Comp. J.,", (1965)

Language and Document Processing

Spanish POS Tagger by Means of Hybrid Methods for the Accentuation of Words

Raymundo Montiel, Blanca E. Pedroza, Ma. Guadalupe Medina

Instituto Tecnológico de Apizaco
División de Estudios de Posgrado e Investigación
Avenida Instituto Tecnológico s/n, C.P. 90300, Apizaco, Tlaxcala México
mlirary@yahoo.com.mx, thelismedina@hotmail.com

Abstract. The task of Word Sense Disambiguation has as aim to identify the correct sense of a word in a context. The resolution of the ambiguity is a complex and useful task for many applications of natural language processing, for example: text categorization, machine translation, restoration of accents and, in general, in information retrieval. The process to accentuate words is also an ambiguity problem, because many words within the Spanish Language that can be marked or not depending on the context in which they are being used. This paper proposes a model that allows to obtain a correct accentuation of words with diacritical accent based on part of the speech tagging by means of the application of hybrid methods (supervised and unsupervised algorithms).

1 Introduction

One of the most difficult tasks and that has reached great interest within the *Natural Language Processing (PLN)*, takes place when several senses or meanings are associated with a word; this phenomenon of language is known as *polysemy*. The task of Word Sense Disambiguation consists of identifying the correct sense of a word in a context [10]. The problem of polysemy is closely related to the problem of the assignment of grammatical categories, which consists of saying if a word is a verb, an article or a noun [6], depending on the meaning that corresponds to that word in agreement to the context of the sentence.

The process of accentuate words is also a problem of ambiguation, because many words within the Spanish that they can be accentuated or not depending to different situations, such as the context, the time of action of the sentences, etc. The diacritical accent allows to distinguish words with identical form, that is, words written with the same letters, but that belong to different grammatical categories.

The lack of accents marks in some words within the sentences, is due to problems of ambiguation. The most common ambiguities in the accentuation of words are met between the words with endings in "o", as is the case of "completo" vs. "completó. They

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 61-68

are the present and past tense, related to verbs with endings in "ar". There are other ambiguities purely semantic, including the nouns: "secretaria" (Person who takes charge writing the correspondence [8]) and "secretaría" (Section of an organization, institution or company[8]).
For this, we propose to develop a computer tool of semantic disambiguation for the Spanish Language, designed for the correct accentuation of words in written texts. This tool is based on the tagging of the sentences. For this goal, we have implemented a hybrid method.

2 Model for tagging of words

The purpose of this work consists of determining if a word with ambiguity on accent has to take an accent mark or has not, which is determined by the context in the one that word is dealing, with help of the assigned tags. For this part, we use a Hidden Markov Model (HMM).
The Models of Markov describe a process of probability which produces a sequence of events or not observable symbols. They are called "hidden" because of a process of underlying probability that is not observable, but it affects the sequence of observed events [7].
A HMM is characterized by a 5-tupla (Q, V, π, A, B), where:
Q is the set of states of the model. Though the states remain hide, they are known previously for the most of the practical applications. For the case of the labeling word, every label would be a state. Generally all the states are connected in such a way that any of them can be reached from any other one in an alone step. The states are labeled as {1, 2, ..., N}, and the current state in time t is denoted as q_t. In the case of the labeling word, we will not speak about the instants of time, but about the positions of every word inside the sentence.
V is the set of the different events that can be observed in each of the states. Each of the individual symbols that a state can emit is denoted as $\{v_1, v_2, ..., v_M\}$. In case of tagging word, M is size of dictionary and every v_k, $1 \leq k \leq M$, is a different word.
$\pi = \{ \pi_i \}$, is distribution of probability of initial state. Therefore,

$$\pi_i = P(q_1 = i), \qquad \pi_i \geq 0, \qquad 1 \leq i \leq N,$$

$$\sum_{i=1}^{N} \pi_i = 1 \tag{1}$$

$A = \{a_{ij}\}$, is distribution of probability of transitions between states, that is to say,

$$a_{ij} = P(q_t = j|q_{t-1} = i) = P(j|i), \qquad 1 \le i,j \le N, \qquad 1 \le t \le T,$$

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad \forall\ i. \tag{2}$$

$B=\{b_j(v_k)\}$, is the distribution of probabilities of observable events, that is to say,

$$b_j(v_k) = P(o_t = v_k|q_t = j) = P(v_k|j),$$

$$b_j(v_k) \ge 0, \quad 1 \le j \le N, \qquad 1 \le k \le M, \qquad 1 \le t \le T \tag{3}$$

$$\sum_{k=1}^{M} b_j(v_k) = 1, \qquad \forall i.$$

3 Description of proposed solution model.

The general model proposed for the solution of the problem is showed in figure 1. The first task of the model is the lexical analysis, which consists of removing the accents of the words in the sentence of entry, thinking that for the calculation of the parameters of the model, words are needed without accents.

In addition, the lexical analyzer identifies and separates the signs of punctuation of the words, and then it identifies if ambiguous words exist inside the sentence of entry to the model and at the same time it indicates the position of every ambiguous word.

To identify the ambiguous words a comparison is done of each one of the words in the sentences, relating them to the words with ambiguity in the dictionary, which was constructed before.

For the phase of tagging in the model of solution (**Fig. 1**), we applying the modified Viterbi Algorithm [14], since all the posible states are not considered, that is to say, all the labels of the set of labels used, but only the most probable labels assigned to each of the ambiguous words.

To calculate the parameters of Hidden Markov Model we conducted a supervised and not supervised training (figure 2). For the phase of supervised training, we use the tagged corpus CONLL, it is collection of news articles by the Agency of News EFE in the year 2000. The parameter's model can be estimated by *maximum likelihood*, from the relative frequencies of appearance of the events in the corpus.

The parameters in matrix A, matrix B, and vector π, there were calculated taking the words without accents, since this is determined by the tags.

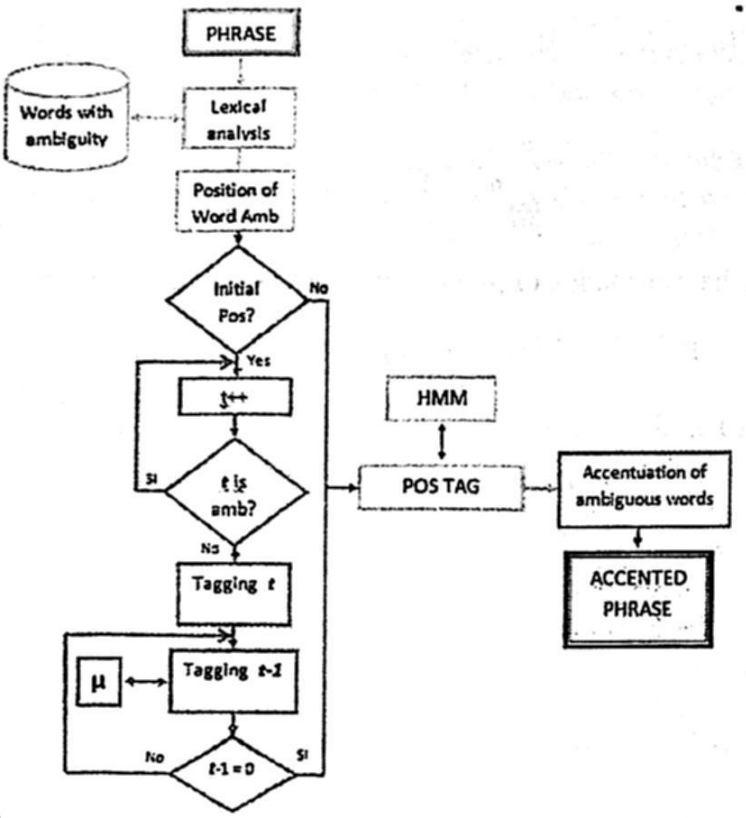


Fig. 1. Diagram of general model of solution proposed

For the matrix A, the probabilities of transition a_{ij} of the equation 2, are obtained counting how many transitions there are from state s_i to state s_j, and dividing by how many transitions there are by state s_i.
For the matrix B, the probabilities of emission, of the equation 3, are obtained counting how many emissions of symbol (v_k) are produced from state (s_j), and dividing by how many times that symbol has passed along by state s_j.
Once we have the initial parameters of μ (equation 1), we appy the algorithm Baum-Welch [1], which increases the probability of the transitions between the states and his symbols, so the probability is improved for the given sequence of observations.

3.1 Example applying the model

Considered the following sentence of entry:

"El jugo frío esta sobre la mesa"

In this sentence several words exist with diacritical accent, even to the beginning of the same one. In this case t words travels until finding a non ambiguous word, "frío". Once opposing the first non ambiguous word, it is assigned the most probable Tag.

After this, the label is assigned but probable t-1 from the already known tag (t) (fig.3), until arriving to t=1.

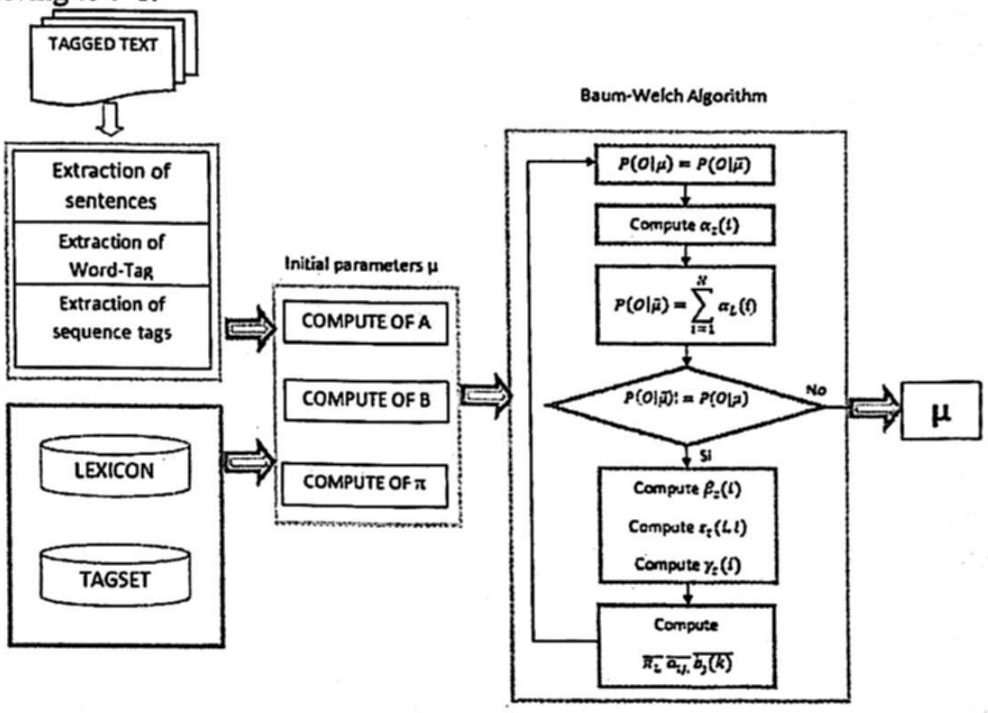


Fig. 2. Supervised and not supervised training

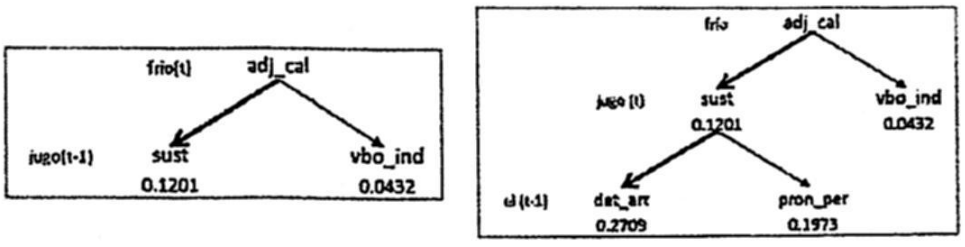


Fig. 2. Allocation of tags

Obtained the first tag of the sentence (if it is the case), the calculation to obtain the tag of the ambiguous words inside the sentence, it is realized only on the tags assigned manually to these words. Table 1 shows the sentence with assigned tag.

Table 1. Result of tagging

El	jugo	frío	esta	sobre	la	mesa
article	noun	adj_qua	verb	prep	article	noun

After having the tags of each one of the words of the sentence, we decide if the ambiguous words will have to or not to take accent, in agreement to our dictionary of ambiguous tagged words (Table 2).

Table 2. Example of ambiguous words together with its tags

Ambiguous word	Tag	Word
El	Article Pronoun	El Él
Esta	Det_dem Verb	Esta Está
jugo	Noun Vbo_ind	Jugo Jugó

Finally, we do a comparison of the words of the sentence together with the assigned tag and we determine if the word has to or not to take accent. Table 3 shows the result of our example.

Table 3. Result of the example.

El	jugo	frío	está	sobre	la	mesa

Besides accentuating the words with diacritical accent, the aim is to accentuate the words that always are accentuated, that is to say those words that must take accent mark. For this a database of accentuated words was realized.

4 Results

For the evaluation of the model, we met a collection of 63 texts of different contexts (Sports, Science, Health, etc.) obtained from the digital newspaper "El Universal[1]", these articles were evaluated by the model from complete phrases. Table 4 shows a summary of the obtained results. Here we have:

1 Number of Text
2 Number of words for Text
3 Number of words with diacritical accent
4 Number of words with diacritical accent, without accent.
5 Number of words with diacritical accent, without accentuating.
6 Number of words with diacritical accent, with accent.
7 Number of words with diacritical accent, accentuated.
8 Number of words without diacritical accent
9 Number of words without diacritical accent, accentuated.

[1] http://www.eluniversal.com.mx/noticias.html

Table 4. Summary of obtained results

1	2	3	4	5	6	7	8	9
1	164	26	26	26	0	0	12	12
7	166	24	22	22	2	2	14	8
10	214	30	26	26	4	3	22	21
19	376	54	49	49	5	3	26	23
24	1180	215	191	170	22	19	125	101
47	322	69	67	66	2	2	26	23
48	384	87	79	78	8	7	23	20
53	898	187	164	162	23	21	78	74
58	114	78	68	63	10	8	34	30
63	165	34	28	26	6	6	21	21
RESUL	22883	4238	3769	3611	472	397	1721	1526

With this information we deduce that:

$$Ambiguous\ words = \frac{4238*100}{22883} = 18.52\%$$

Besides these results, the information in the table 5 is evaluated using the metric of efficiency and obtaining the percentages of Recall (r), Precision (p), Mistake (m), Accuracy (a), and F_m.

Table 5. Results

Based on accented words		Based on non-accented words
r = .7153	m = 0.0549	r = .9796
p = .8411	a = .945	p = .9580
f-measure = .7731		f-measure = .9687

In the first part of the table 5, we show the results obtained from the words accentuated by our model, where the *precision* (number of words accentuated divided by the total number of words with diacritical accent) takes 84.11% as a result.

We have identified that one of the problems is in the tagging of the word "el", due to the fact that if the system finds this word at the beginning of the sentence, it assigns to the word the tag of "personal pronoun" and accentuates it, nevertheless, also it can take the tag of "article", in which case does not take accent.

As we mentioned previously, this model also determines when a word with diacritical accent must not take orthographic accent. The second part of the table 5 shows the results obtained from the words that must not take orthographic accent, in this case the *precision* (number of words that the model decide not to accentuate, divided by the total number of words with diacritical accent) takes 95.8% as a result.

Nevertheless, bearing in mind the result of *accuracy* (number of correct decisions done by the model) is 94.5%, having a mistake of 5.49%.

5 Conclusions

Tagging of words is a technique that can help to identify ambiguity in the accentuation of words of a certain text and it can generate computational tools that can help to the correct accentuation of texts in Spanish, being useful to make easier the writing of documents in Spanish or to use it as an assistant to teach grammatical rules.

References

1. Baum, L. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat.* vol. 37, 1554-1563, 1966.
2. Bobiceva, V. (2008). O altă metodă de restabilire a semnelor diacritice. In Pistol I., Cristea D. Tufiş D. (eds.): Resurse Lingvistice şi Instrumente pentru Prelucrarea Limbii Române, pp. 179-188.
3. Dempster A., Laird, N., et al. Maximum likelihood from incomplete data via the EM algorithm. Journal of Royal Statistical Society. Series B (Methodological), vol. 39, No. 1, 1-38, 1977.
4. Marty, F. (1992). Trois systémes informatiques de transcription phonétique et graphémique. En Le Francais Moderne, pp. 179-197.
5. Mihalcea, R. (2002). Diacritics Restoration: Learning from Letters versus Learning from Words. In Proceedings of CICLing, pp. 339-348.
6. Perea Saedón, José Ignacio. (2005). Etiquetado de textos y su aplicación a la traducción. Unplublished monitired research. University of Granada.
7. Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. IEEE, vol. 77, no. 2, pp. 257-286.
8. REAL ACADEMIA ESPAÑOLA: Banco de datos (CORDE) [en línea]. Corpus diacrónico del español. <http://www.rae.es> [29 Abril 2009]
9. Rivest, R. L., Learning decision list, Machine Learning, 2, 229-246, 1987.
10. Stevenson, M; Y. Wilks. Combining independent knowledge sources for word sense disambiguation. En R. Mitkov (ed), Recent Advances in Natural Language Processing, John Benjamins Publisher, 2000.
11. Wagacha, P., De Pauw, G., et al. (2006). A Grapheme-Based approach for accent restoration in Gikuyu. In Proceedings of the Fifth International Conference on Language Resources and Evaluation, pp. 1937 – 1940.
12. Yarowsky, D. Decision list for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In Proceeding of the 32th Annual Meeting of the Association for Computational Linguistics, (ACL'94). 1994. 88-95.
13. Yarowsky, D. A comparison of corpus-based techniques for restoring accents in Spanish and French text. To appear in Proceedings, 2nd annual Wordshop on Very Large Text Corpora, Kyoto, Japan. 1994
14. Viterbi, A. (2006). A personal history of the Viterbi algorithm. Signal Processing Magazine, IEEE, vol. 23, no. 4, pp. 120-142.

A Methodology for Dialogue Act Modeling from Part-of-Speech Annotations of Practical Dialogues in Mexican Spanish

Sergio R. Coria[1] and Luis A. Pineda[2]

[1] Software Engineering Group, University of the Sierra Sur, Calle Guillermo Rojas Mijangos S/N, Col. Ciudad Universitaria
70800 Miahuatlan de P. Diaz, Oax., Mexico
coria@unsis.edu.mx
[2] Department of Computer Science, National Autonomous University of Mexico, Circuito Escolar S/N, Ciudad Universitaria
04510 Mexico, D.F.
luis@leibniz.iimas.unam.mx

Abstract. This article proposes a methodology to model dialogue acts (DA) from part-of-speech (POS) annotations in practical dialogues in Mexican Spanish. An example using empirical data is presented to illustrate this methodology. DA and POS annotations were obtained from DIME, a spoken dialogue corpus in which speakers perform tasks in a virtual environment on a collaborative basis. POS structure is a key notion in this methodology. It is the POS tagging sequence for particular instances of utterances. The models are produced as probability tables of pairs of DA annotations and POS structures. Results show that a number of POS structures are more frequently used by speakers to communicate particular DA types. These patterns can be exploited to enhance systems for automatic recognition of DA.

Keywords: spoken dialogue, dialogue act modeling, DAMSL, DIME-DAMSL, part of speech.

1 Introduction

Practical dialogue, also known as task-oriented conversation, is that in which speakers cooperate to achieve a common goal. This type of conversation is simpler than general conversation because the number and complexity of its dialogue act (DA) types are less. In addition, it is the typical conversation to be performed in automatic systems for dialogue management. An advantage for computational purposes is that its analysis and modeling can be better controlled on experimental conditions.

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 69-76

The relation between DA type and POS in Spanish has been poorly investigated from an empirical, computational, view. Therefore, this research aims to propose a methodology to characterize its patterns in spoken dialogue corpora. Other motivation for this methodology is to improve the performance of dialogue management systems by taking advantage of a number of information sources from speech in addition to lexical content.

This paper is organized as follows: section 2 describes a previous research in the area. Section 3 comments theoretical and computational foundations of DA from the views of DAMSL and DIME-DAMSL annotation schemes. Section 4 addresses part of speech (POS) annotation. Section 5 describes a methodology for modeling DA from POS on an empirical, statistical, basis; an example using data from the DIME corpus is explained. Section 6 presents an overview of the DIME corpus and the data sample for the example. Section 7 presents the example results. Finally, section 8 discusses the results and suggests future research.

2 Previous Research in the Area

Instances of previous research with a similar approach are: [1], [2], [3], [4] and [5]. Most of them address the problem for English specifically. Specific research for Spanish is scarce, so this is a strong motivation for addressing the issue. [1] and others address the problem on subsets of DA types only, *e.g.* [3] studies the specific domain of business-appointment scheduling. Others use high complexity methods, such as Hidden Markov Models. [2] uses statistical language models produced from word transcription or automatic recognition. [4] presents a supervised adaptation method for dialog act tagging, evaluating model adaptation for dialog act tagging by using out-of-domain data or models. [5] explores the tasks of dialogue act segmentation and classification by employing simple lexical and prosodic knowledge sources. [6] addresses the problem from the perspective of intonational information in Spanish and it uses the same empirical resource that the present research uses. Unlike most of previous work, the present research proposes a methodology that creates models as probability tables on POS sequences and DA annotation pairs.

3 Dialogue Act

Searle's speech act is the production or emission of an utterance-instance under certain conditions, and speech acts are the basic or minimum units of linguistic communication. On this basis, DA is the characterization of a speech act within the context of a task-oriented conversation. From a computational view, DA need to be represented so that they can be analyzed within experimental conditions. Therefore, a number of DA

representation schemes, construed by annotation tag sets and annotation rules, have been proposed. A widely used scheme is DAMSL [7]. This research uses DIME-DAMSL [8], a scheme that is based on DAMSL and that provides with high inter-annotator consistency. DIME-DAMSL identifies two communication planes in DA: *obligations* and *common ground*. Tables 1 and 2 present the tag set of DIME-DAMSL scheme. A compound tag involves that an utterance communicates two or more DA simultaneously.

Table 1. Simple and compound tags for annotation of obligations DA (from DIME-DAMSL).

info-request	*answer*	*no-tag*	*action-dir*
commit	*info-request answer*	*action-dir answer*	*offer*
info-request offer	*action-dir offer*		

Obligations DA are those in which an obligation to perform an action or to provide some piece of information is generated either on the listener or on the speaker.

Table 2. Simple and compound tags for annotation of common ground DA (from DIME-DAMSL).

no-tag	*accept*	*affirm*	*hold repeat-rephr*
open-option	*accept-part*	*reaffirm*	*hold*
reject	*affirm accept*	*ack*	*NUS(non-understanding signal)*
offer conv-open	*repeat-rephr*	*reject-part*	*conv-close*
affirm-reject	*open-option accept*	*offer*	*accept hold repeat-rephr*
affirm maybe	*affirm hold*	*offer accept*	*affirm conv-close*
affirm accept exclam	*affirm accept-part exclam*	*affirm correct*	*affirm perform conv-close*
hold NUS	*open-option reject*	*perform*	*reaffirm complement*
reaffirm hold	*hold in task-management*	*other*	

Common ground DA are those in which shared knowledge or beliefs are established or re-established (*agreement* subplane). Also, these DA allow the dialogue participants to manage the communication channel (*understanding* subplane).

Table 3. Tags for POS annotation (reproduced from [9]).

Tag	Description	Tag	Description	Tag	Description	Tag	Description
A	Adjective	*AD*	Demonstrative adjective	*C*	Conjunction	*N*	Noun
P	Pronoun	*PC*	Clitic pronoun	*PI*	Interrogative pronoun	*PR*	Relative pronoun
R	Adverb	*RA*	Acceptation adverb	*RI*	Interrogative adverb	*RN*	Negation adverb
RR	Relative adverb	*S*	Preposition	*TD*	Definite article	*TI*	Indefinite article
V	Verb	*VAM*	Modal auxiliary verb	*VC*	Verb with clitic		

4 Part of Speech (POS)

Part of speech (POS) is the characterization of a word according to the function that it performs in an utterance. A POS annotation process is based on a POS tag set and annotation rules that guide a human or automatic annotator to assign a label to a particular

instance of a word in an utterance. A number of POS tag sets and annotation rules exist. This research uses [9], which is reproduced in Table 3.

Table 4. POS structures and their frequencies in the dataset.

POS structure	Frequency	% (threshold=0.6%)	Accum. %
RA	170	20.0	20.0
R_V_R	115	13.5	33.5
R	20	2.4	35.8
PD	14	1.6	37.5
RN	13	1.5	39.0
RI_V_C_PC_V	9	1.1	40.1
V_TI_N	7	0.8	40.9
RI_V_R	6	0.7	41.6
Other structures (each <= 0.6%)	497	58.4	100.0
Total	851		

5 Methodology

The methodology proposed by this research to modeling DA from POS annotations is based on correlation analyses of data sets from both DA and POS annotations in a spoken dialogue corpus. Models are construed on a Pareto analysis basis and probability tables and they can also be represented as classification trees.

POS structure is the basic notion in this methodology. POS structure is the complete sequence of POS labels that are assigned to words in a particular utterance. The whole set of POS structures and the probabilities of DA types for each POS structure in a corpus are a simplified statistical language model of DA types obtained from that specific corpus. Once POS annotations of the corpus are available, the basic steps to perform are:

Step 1. Creation of POS structures from POS label sequences: this involves concatenating the POS labels of each utterance by inserting a character such as _ (*underscore*) or any other similar to produce one single string. For instance, a POS sequence as *R, V, R* is transformed into *R_V_R*.

Step 2. Pareto analysis of POS structures: its purpose is to identify the most frequent POS structures (see Table 4). This is needed to determine a threshold for discarding POS structures with lowest frequencies. These structures might be useless for DA modeling because they are associated to syntactic structures that are scarcely used by speakers. The threshold can be determined by an empirical fine-tuning for specific applications. Every POS structure with a relative frequency less than the threshold cannot be recognized by the model; however, this behavior is useful because the recognition task can be focused on the most frequent structures.

Step 3. Statistical analyses of DA for each of the most frequent POS structures for obligations and for common ground tags (see Tables 5 and 6). Two separated analyses (*i.e.* one for obligations and one for common ground) are needed because DIME-DAMSL scheme assigns both an obligations and a common ground tag for every single utterance.

In a real-world application, DA recognition can be supported by mapping a particular POS structure to its most probable DA types on obligations and common ground. Therefore, statistical analysis of these patterns is the core of a model. A threshold for relative frequencies of DA tags that are associated to every POS structure is convenient to disregard DA types that are not statistically significant in the dataset.

Table 5. Obligations tags of the most frequent POS structures.

POS structure	% of dataset (threshold=0.6)	Obligations tag	% of POS structure (threshold=30.0)
RA	20.0	*answer*	52.4
		Other	47.6
R_V_R	13.5	*info-request*	82.6
		Other	17.4
R	2.4	*no-tag*	40.0
		answer	35.0
		Other	25.0
PD	1.6	*info-request*	71.4
		Other	28.6
RN	1.5	*answer*	53.8
		no-tag	30.8
		Other	15.4
RI V C PC V	1.1	*info-request*	100.0
V_TI_N	0.8	*no-tag*	42.9
		Other	57.1
RI V R	0.7	*info-request*	100.0
Other structures (each <= 0.6%)	58.4		

Step 4. Statistical analyses of the most frequent POS structures for each DA type. Like analyses in step 3, two analyses (one for obligations and one for common ground tags) are produced. The purpose is to identify the POS structures that occur most frequently for each DA type. Usually, only a lower number of pairs of POS structure and DA type with high percent for the pair exist in a corpus. The identification of these patterns is useful to validate results from steps 2 and 3. See Table 7 for obligations. Also, a table for common ground is available at the WWW[1].

Step 5. Implementation of two DA recognition models (one for obligations and one for common ground). Models can be implemented in a real-world dialogue management system considering the following stages: 1) word-level automatic speech recognition that parses words into POS, 2) POS tagging can then be transformed into POS structures, and 3) the DA tag with highest probability is assigned for the POS structure using the tables previously computed. One tag is assigned for obligations and one for common ground. Thresholds defined in steps 2, 3 and 4 involve that the recognition task is focused on the most frequent DA types and POS structures.

This methodology is illustrated by an example on a practical dialogue corpus that is explained below.

[1] http://www.unsis.edu.mx/~coria/mwpr_2009/table_7_A.pdf

Table 6. Common ground tags of the most frequent POS structures.

POS structure	% of dataset (threshold=0.6)	Common ground tag	% of POS structure (threshold=30.0)
RA	20.0	*accept*	94.1
		Other	5.9
R_V_R	13.5	*no-tag*	86.1
		Other	6.2
R	2.4	*accept*	40.0
		Other	60.0
PD	1.6	*hold_repeat-rephr*	64.3
		Other	35.7
RN	1.5	*reject*	46.2
		Other	53.8
RI_V_C_PC_V	1.1	*no-tag*	55.6
		accept	33.3
		Other	11.1
V_TI_N	0.8	*no-tag*	42.9
		Other	57.1
RI V R	0.7	****	****
Other structures (each <= 0.6%)	58.4		

6 Empirical Resource

The DIME Corpus [10] is the empirical resource used in this research. An empirical approach is preferable for DA modeling because it increases the model efficiency and allows creating probabilistic, instead of deterministic, models. The DIME corpus contains 26 screen videos and audios of two-person practical dialogues. In each dialogue one person plays the role of the computer *System* and his partner acts as the system *User*. Videos contain the graphical status of a virtual scenario and graphical actions performed by individuals on virtual pieces of furniture as they interact in a CAD (computer aided design) software. The common goal for participants in every dialogue is to arrange pieces of furniture in a virtual kitchen while satisfying a number of design specifications and constraints. The corpus also contains orthographic, phonetic, POS and DA transcriptions, among others. A sample containing 851 utterances from 12 dialogues is used for this research. The sample includes the orthographic, POS and DA annotations of utterances.

7 Example Results

Results from an example to illustrate the methodology using data from the DIME corpus are presented in Tables 4 to 7.

Data in Table 5 are ordered by POS structure percent (second column). Percents of DA tags are ordered for each POS structure. A useful alternate ordering can be by percent of

majority class of POS structure; *i.e.* the pairs *RI_V_C_PC_V* with *info-request* and *RI_V_R* with *info-request* would be the top 2, each with 100.0%, then *R_V_R* with *info-request* with 82.6% would be third, etc. This way, the most significant patterns are easily identified. Table 6 is similar, but its values for *RI_V_R* do not satisfy the threshold. The complete models, *i.e.* with no thresholds, are available at the WWW[2].

Table 7. Most frequent POS structures for obligations DA types.

Obligations tag (single or compound)	Frequency	%	Most frequent POS structure	% of obligs. tag (threshold=35.0)
info-request	262	30.8	*R_V_R*	36.3
			Other	63.7
answer	213	25.0	*RA*	41.8
			Other	58.2
commit	34	4.0	*RA*	91.2
			Other	8.8
info-request_answer	16	1.9	*R_V_R*	43.8
			Other	56.2

8 Discussion and Future Research

A methodology for DA modeling from POS annotations has been presented. POS structure, *i.e.* the sequence of POS tags from a particular utterance, is an important notion in this methodology. Results from an empirical analysis show that a number of POS structures are more frequently used by speakers to communicate specific DA types in practical dialogues. Some examples of the most frequent pairs are: *RA* for *commits*, *R_V_R* for *info-requests*, *RN* for *reject-parts*, etc.

Defining thresholds for the most frequent DA types and/or POS structures is a solution to identify the most significant patterns. Therefore, this approach can focus the recognition task on the DA types and/or the POS structures that have highest frequencies in a training corpus.

The main contributions and significance of this research are: 1) results suggest that a number of DA types are uttered using specific subsets of POS structures more frequently than other structures and these patterns could be exploited to improve DA managements systems, 2) the methodology is simple and it can be used on other languages for theoretical and practical purposes, 3) although patterns that are discovered from a specific corpus cannot be used to improve general-purpose DA management systems, the methodology can be useful on a domain-specific basis. Implementations for real-world systems should use other additional sources from speech, such as intonation, cue words, etc. to obtain higher accuracy rates, since only using POS structure of utterances does not suffice.

[2] See obligations and common ground models at http://www.unsis.edu.mx/~coria/mwpr_2009/

Future research can address applying the methodology to implement and evaluate an automatic recognition model for DA using standard evaluation metrics. Also, comparing results to similar systems is needed. Implementation can be based on probability tables for the pair patterns. In a later implementation, other models can include POS structure information along with intonation and other sources. The models can be implemented in decision trees, such as those created with classification and regression algorithms.

Acknowledgments. The authors thank the DIME Project team for their effort in producing POS and DA annotations of the DIME Corpus.

References

1. Jurafsky, D., Shriberg, E., Fox, B., Curl, T., 1998. Lexical, prosodic, and syntactic cues for dialog acts. In: Procs. of the ACL/COLING Workshop on Discourse Relations and Discourse Markers, Montreal, Canada, August 1998, pp. 114--120.
2. Shriberg, E., Bates, R., Stolcke, A., Taylor, P., Jurafsky, D., Ries, K., Coccaro, N., Martin, R., Meteer, M., Van EssDykema, C., 1998. Can prosody aid the automatic classification of dialog acts in conversational speech? USA, Language and Speech 41(3–4) 439-487 (Special Issue on Prosody and Conversation).
3. Jekat, S., Klein, A., Maier E., Maleck, I., Mast, M., Quantz, J. J., 1995. Dialogue Acts in VERBMOBIL, VM-Report 65.
4. Tur, G., Guz, U., Hakkani-Tür, D., 2006. Model adaptation for dialog act tagging. In: Proceedings of SLT 2006, 1st biannual IEEE/ACL Workshop on Spoken Language Technologies, Aruba, December 2006.
5. Ang, J., Liu, Y., Shriberg, E., 2005. Automatic dialog act segmentation and classification in multiparty meetings. In: Proceedings of ICASSP.
6. Coria, S.R., Pineda, L.A. An Analysis of Prosodic Information for the Recognition of Dialogue acts in a Multimodal Corpus in Mexican Spanish. Computer Speech and Language, Vol. 23 (2009), pp. 277--310, Elsevier.
7. Allen, J.F., Core, M., 1997. Draft of DAMSL: Dialogue Act Markup in Several Layers. Technical Report, The Multiparty Discourse Group. University of Rochester, Rochester, USA.
8. Pineda, L.A., Estrada, V.M., Coria, S.R., 2006. The obligations and common ground structures of task oriented conversations. In: Proceedings of the Fourth Workshop in Information and Language Technology TIL-2006, in IBERAMIA 06, Brazil.
9. Moreno, I., Pineda, L., Speech Repairs in the DIME Corpus, Research in Computing Science, Vol. 20, pp. 63 – 74, 2006.
10. Villaseñor, L., Massé, A., Pineda, L. The DIME Corpus, Memorias 3er. Encuentro Internacional de Ciencias de la Computación ENC01, Tomo II, C. Zozaya, M. Mejía, P. Noriega y A. Sánchez (eds.), SMCC, Aguascalientes, Ags. México, Septiembre, 2001.

Feature Selection on the Web People Search Task*

David Pinto, Mireya Tovar, Beatriz Beltrán, Darnes Vilariño, Héctor Furlog

Faculty of Computer Science, BUAP
14 Sur & Av. San Claudio, CU, Edif. 104C
Puebla, Mexico, 72570
{dpinto, mtovar, bbeltran, darnes}@cs.buap.mx
http://nlp.cs.buap.mx

Abstract. Searching people on the Web is one of the most common activities carried out by Internet users. However, search engine results are usually generated without taking into account the inherent ambiguity of people names. Homonymy is the state of one of a group of words of sharing the same spelling and the same pronunciation but having different meanings. In this paper we have focused our research on the task of discriminating people that share the same name but have different occupation. We compare two different approaches for selecting features on documents of homonyms from a supervised and unsupervised viewpoint. We have used a high scalable clustering method based on fingerprinting in order to discriminate a set of homonyms taken from a testbed corpus used in one international competition. The proposed system performed well in comparison with other results reported in literature.

1 Introduction

The homonym discrimination on the web is a task that requires a special attention by the natural language engineering community. Searching people in Internet is one of the most common activities performed by the World Wide Web users [1]. The main challenge consists on discriminating people with presence in the Web that share the same name but have different occupation. In Figure 1 we can see a snapshot of Google[1] when searching people on the Web. The results retrieved are neither considered in any way belonging to people nor to be classified or clustered. This an undesirable behaviour of a wise search engine.

Spock.com[2] instead assumes by default that the user is searching people on Internet and, therefore, it brings together all the results (people webpages) that the system considers that share the same occupation (see Figure 2). In theory, an information retrieval system which is able to recognize queries related to people should have similar behaviour to the Spock system; otherwise, it should have a google-like behaviour.

* This work has been partially supported by the CONACYT project #106625, as well as by the PROMEP/103.5/09/4213 grant.

[1] http://www.google.com

[2] http://www.spock.com

© J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad and H. Sossa (Eds.)
Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 77-85

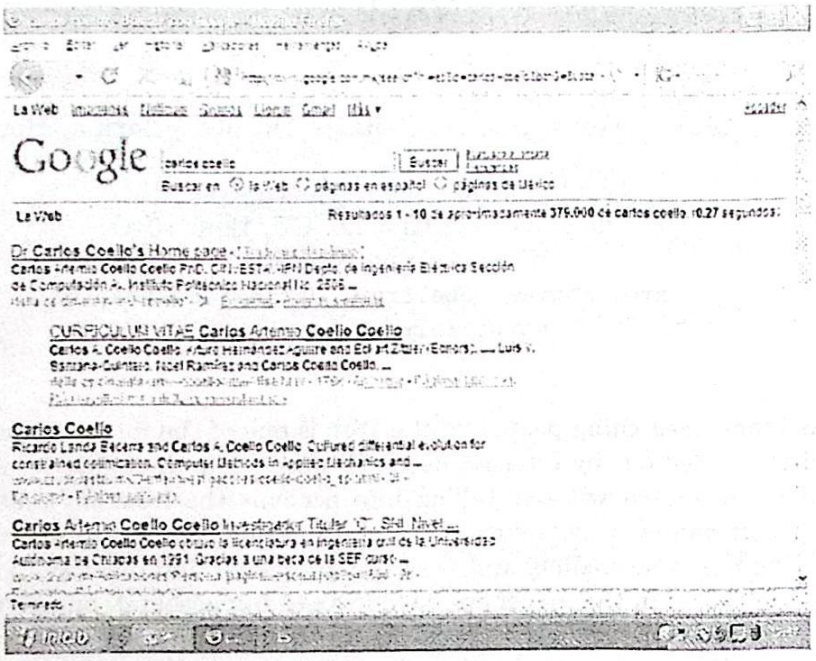


Fig. 1. Google results when searching people on the WWW.

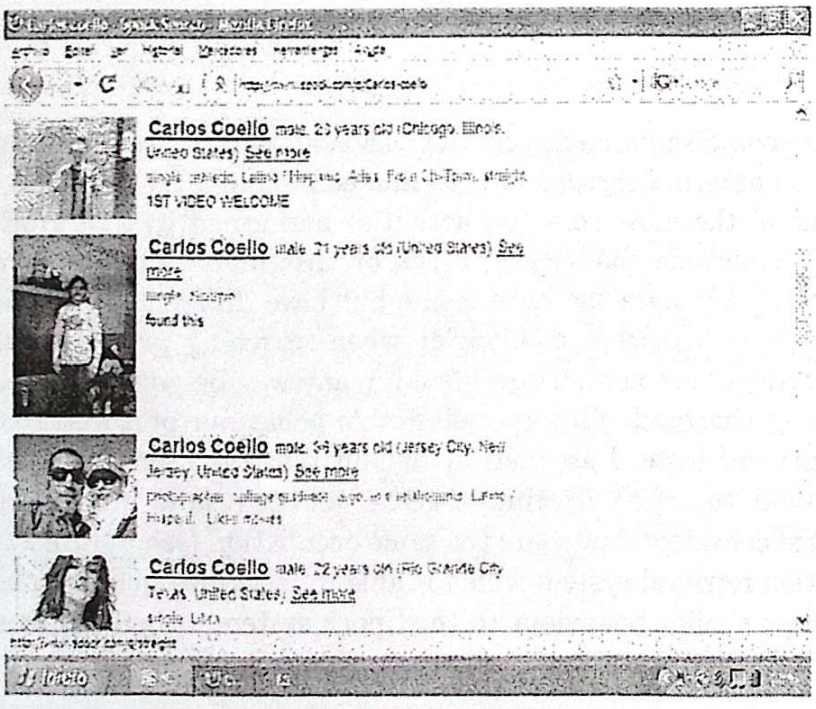


Fig. 2. Spock results when searching people on the WWW.

The aim of this paper is to automatically discover the most salient features for the classification of homonyms. We present a semi-supervised classification method which relies the training phase on the WePS-1 collection [1], which contains 49 ambiguous people names. We evaluate the obtained features on a test dataset in order to see the performance obtained on this task. Moreover, we report the performance of the classifier when using the WePS-2 collection, which is a standard corpus used at the Web People Search Task [2] made up of 30 ambiguous names of people, each name with a number of HTML pages with information related with that people name. The complete description of the evaluated corpus is given into detail in [2].

The classifier used on the experiments was constructed on the basis of a system which relies its text retrieval techniques on hash functions [3]. In particular, we have constructed a new vectorial coordinate system for the representation of the original data and, thereafter, we calculate the distance of the vectorial representation of each input dataset by means of a hash function.

We must take into account that the fingerprinting technique may allow indexing and clasifying of documents in a one single step. Therefore, given the huge amount of information available in Internet, we consider that an important contribution of this research work consists of providing a very fast way of classifying people names on the World Wide Web.

The evaluation of the experiments carried out show that the implemented technique could have a positive impact in the analysis/indexing of huge volumes of information. However, the feature set for all the documents in the WePS framework needs to be further investigated.

The remainder of this document is structured as follows. In Section 2 we describe the components of the implemented system. Section 3 describes into detail the two feature selection techniques used in the experiments. The document fingerprint technique used in the document indexing and clustering process in the Web People Search framework is explained in Section 4. The experimental results are discussed in Section 5, whereas the conclusions are given in Section 6.

2 Description of the implemented system

The system architecture follows the classical approach of supervised classification (see Figure 3) and it comprises the following components:

Pre-processing: We have programmed two implementations in order to perform the HTML to text conversion. The first HTML to text converter was programmed with Java, whereas the second was implemented with AWK. No HTML tags nor url's were considered in the text extraction.

Feature selection: The process of document feature selection is performed in two steps as presented in Figure 4, and it is described in Section 3. In a brief, all the categories of different people names are clustered in order to conform a general set of features which hopefully will describe a certain occupation.

Document representation: We see each document as a vector of features and, therefore, we may formally express the representation of each webpage as follows.

$$\vec{d} = [w(f_1), w(f_2), \cdots, w(f_n)], \quad (1)$$

where $w(f_i)$ is the weigth of the i-th feature recognized in the document d.

Indexing/clustering: The indexing process was carried out by using the formula expressed in Equation (5). We used a specific threshold (ϵ) in order to determine a range of hash-based values (documents) that should belong to the same cluster. The overlapping of clusters was not considered but it may be easily implemented.

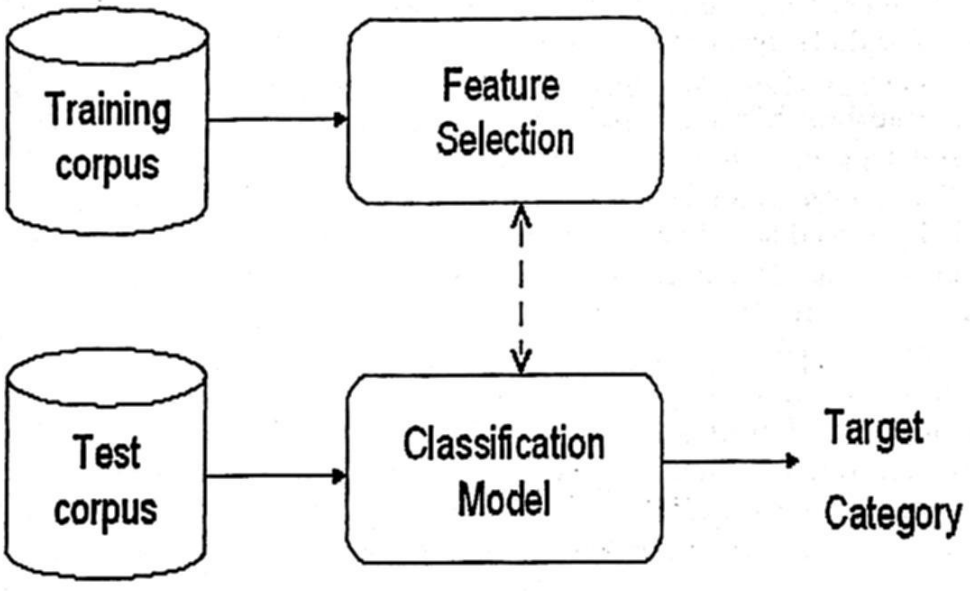


Fig. 3. The classification approach.

3 Feature Selection Techniques

We have considered that each person webpage should be classified according to the abilities of that person with respect to a specific occupation. Therefore, we focus our investigation on discovering such features that identify the role of a given person on his occupation. However, we do not expect that all the occupations will be represented on the training corpus nor that each person will completely describe the features of some occupation.

Our approach extract the most salient features of each occupation. This process is carried out within all the person webpages by means of the two following feature selection techniques.

Entropy: We calculate the entropy (H) of each word (w_i) within each cluster/category for every person. Formally, given a cluster (occupation) of one person C_j, which is made up of a document set ($\{d_1, d_2, \cdots, d_n\}$). The most salient features of C_j are those words w_i that are obtained by Equation (2).

$$Features(C_j) = \{w_i | H(w_i, C_j) > \beta\} \tag{2}$$

with

$$H(w_i, C_j) = p(w_i, C_j) * log(p(w_i, C_j)) \tag{3}$$

where $p(w_i, C_j)$ is the probability of word w_i in cluster C_j, and β is a real value which is used as a threshold, whose value range between cero and one.

Conditional probability: We selected those words that represent better each category by using the conditional probability of word w_i given the category C_j, i.e., $p(w_i|C_j)$. We sort these probabilities and, thereafter, we just selected those that are above of a given threshold γ ($\gamma \in \mathbb{R}$ and $\gamma \in [0, 1]$) as shown in Equation (4).

$$Features(C_j) = \{w_i | p(w_i|C_j) > \gamma\} \tag{4}$$

Once the most salient features of each category are obtained, we cluster these features in order to bring together all that features that belong to the same occupation by using the K-Star clustering method [4]. In summary, given the k categories we are using for extracting the most salient features, we firstly extract k sets of features (one per category) and, thereafter, we consider merging these feature sets by using a clustering method in order to get k' new feature sets ($k' \leq k$). The complete process is presented in Figure 4.

4 Document Fingerprinting and classification

Document indexing based on fingerprinting is a powerful technology for similarity search in huge volumes of documents. The goal is to provide a proper hash function which quasi-uniquely identifies each document, so that the hash collisions may be interpreted as similarity indication.

4.1 Document classification

Formally, given two documents d_1 and d_2, and the fingerprint of the two documents $h(d_1)$ and $h(d_2)$, respectively. We consider d_1 and d_2 to be ϵ-similar iff $|h(d_1) - h(d_2)| < \epsilon$.

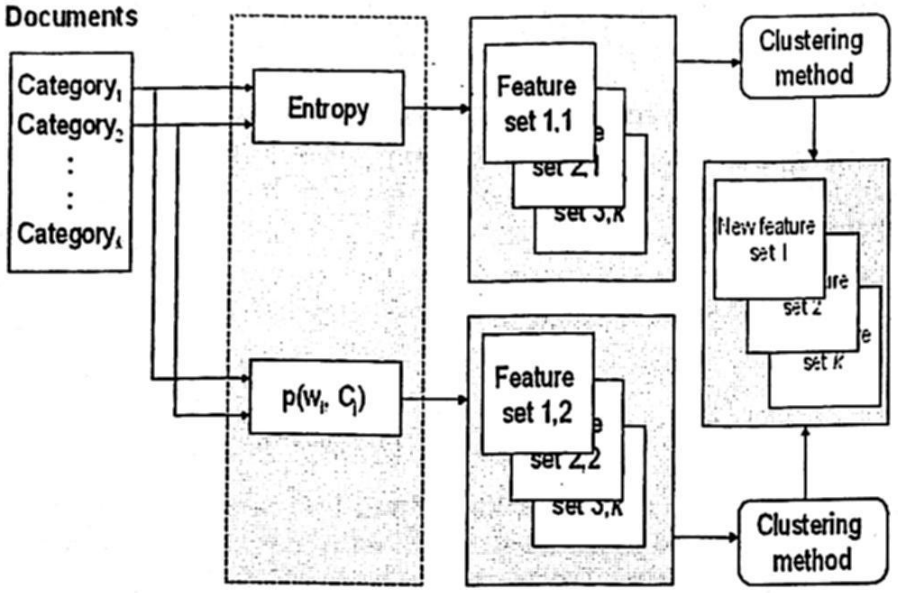


Fig. 4. The process of discovering general features for the homonymy descrimination task.

4.2 Document fingerprinting

In the context of document indexing/clustering/retrieval a fingerprint $h(d)$ of a document d may be considered as a set of encoded substrings taken from d, which serve to identify d uniquely.

Defining the specific hash function to encode the substrings of the documents is the main challenge of the fingerprinting technique. In particular, in the implementation of the proposed Web People Search system we defined a small set k of term-frequency vectors ($\{\vec{r_1}, \vec{r_2}, \cdots, \vec{r_k}\}$) (which are used as reference for a new coordinate system) in order to be considered as the new reference for the vectorial representation of each document of the WePS-2 collection. In Figure 5 we may see an overview of the proposed approach.

Formally, given a set of k reference vectors, $\{\vec{r_1}, \vec{r_2}, \cdots, \vec{r_k}\}$, and the vectorial representation of a document d ($\vec{d}$). We defined the fingerprint of $\vec{d}$ as shown in equation (5).

$$h(\vec{d}) = \sum_{i=1}^{k} \vec{r_1} \cdot \vec{d}^{\,t} \tag{5}$$

The specific features used in the vectorial representation of the documents are the ones explained in Section 3.

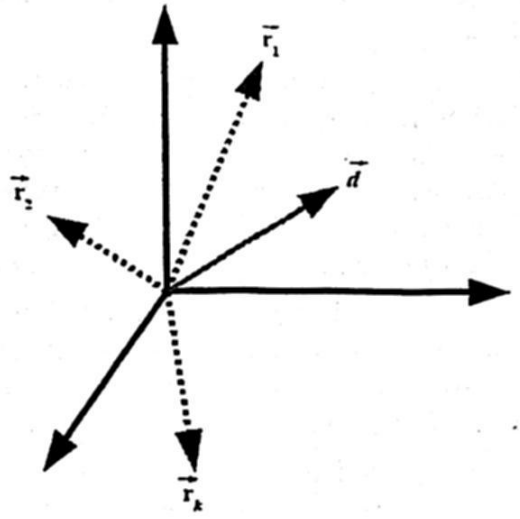


Fig. 5. The new vectorial coordinate system used on the implemented hash-based function for fingerprinting.

5 Experimental results

Besides the evaluation of the WePS-2 collection, we performed a set of experiments over the training and test dataset of the WePS-1 collection (see [1] for a complete description of these datasets). The obtained results are presented in Tables 1, 2 and 3, respectively.

In these tables we may see the following set of metrics used to evaluate the performance of the implemented system:

BEP: BCubed Precision
BER: BCubed Recall
FMeasure_0.5_BEP-BER: F-measure of B-Cubed P/R with alpha set to 0.5
FMeasure_0.2_BEP-BER: F-measure of B-Cubed P/R with alpha set to 0.2
P: Purity
IP: Inverse Purity
FMesure_0.5_P-IP: F-measure of Purity and Inverse Purity with alpha set to 0.5
FMeasure_0.2_P-IP: F-measure of Purity and Inverse Purity with alpha set to 0.2

For more details about the evaluation metrics please refer to [5]. The baselines and the rationale for F-measures with alpha 0.2 are explained in the WePS-1 task description paper [1].

We have tested the two approaches that result from selecting features by using entropy and conditional probability on each category. We may see (Tables 1, 2 and 3) that the implemented approaches obtained a performance comparable

with two of the proposed baselines, ALL_IN_ONE and ONE_IN_ONE, with a document similarity threshold (ϵ) equal to 0.4.

Although, some of the implemented approaches obtained acceptable results in comparison with the baselines, in the case of the WePS-1 test collection, we could not outperform one of the proposed baselines. We consider that the expected document distribution over the final clusters has played an important role on the obtained results, since the presented algorithm of fingerprinting usually assumes a uniform distribution of documents over the discovered clusters.

The evaluation of the experiments carried out shows that the implemented technique could have a positive impact in the analysis/indexing of huge volumes of information. However, the feature set for all the documents in the WePS framework needs to be further investigated.

As future work, we would like to experiment on feature selection in order to clearly benefit the construction of the reference vector set. We are considering the use of other supervised classifiers in order to extract the most important features of the WePS collection.

Finally, we would like to analyse the use of new hash-based functions and new document representations which consider characteristics other than those based on term frequencies.

Table 1. Evaluation of the WePS-2 test dataset.

run	BEP	BER	FMeasure_0.5 BEP-BER	FMeasure_0.5 P-IP	IP	P
ALL_IN_ONE_BASELINE	0.43	1.0	0.53	0.67	1.0	0.56
COMBINED_BASELINE	0.43	1.0	0.52	0.87	1.0	0.78
ONE_IN_ONE_BASELINE	1.0	0.24	0.34	0.34	0.24	1.0
Entropy	0,46	0,9	0,54	0,65	0,94	0,57
Conditional probability	0.44	1.00	0.53	0.67	1.00	0.56

Table 2. Experimental results with the training dataset of the WePS-1 collection.

run	BEP	BER	FMeasure_0.5 BEP-BER	FMeasure_0.5 P-IP	IP	P
ALL_IN_ONE_BASELINE	0.54	1.0	0.64	0.75	1.0	0.65
ONE_IN_ONE_BASELINE	1.0	0.34	0.45	0.46	0.35	1.0
COMBINED_BASELINE	0.48	1.0	0.60	0.9	1.0	0.82
Entropy	0.57	0.93	0.65	0.61	0.96	0.50
Conditional probability	0.54	0.99	0.65	0.61	1.00	0.48

Table 3. Experimental results with the test dataset of the WePS-1 collection.

run	BEP	BER	FMeasure_0.5 BEP-BER	FMeasure_0.5 P-IP	IP	P
ALL_IN_ONE_BASELINE	0.18	0.98	0.25	0.4	1.0	0.29
COMBINED_BASELINE	0.17	0.99	0.24	0.78	1.0	0.64
ONE_IN_ONE_BASELINE	1.0	0.43	0.57	0.61	0.47	1.0
Entropy	0.21	0.92	0.30	0.36	0.96	0.25
Conditional probability	0.18	0.98	0.25	0.36	1.00	0.25

6 Conclusions

We implemented a hash-based function in order to uniquely identify each document from a text collection in the framework of the Web People Search task. The hash collisions were interpreted as similarity degree among the target documents. In this way, we constructed an algorithm which only takes into account the local features of each document in order to index/cluster them. The experimental results over the WePS-1 test and training datasets showed an acceptable performance of the proposed algorithm. However, the proposed reference vector for the fingerprinting-based model did not help too much when evaluating with the WePS datasets. The proper construction of reference vectors for the automatic and unsupervised classification of people names in the Web needs to be further investigated.

References

1. Artiles, J., Gonzalo, J., Sekine, S.: The SemEval-2007 WePS evaluation: Establishing a benchmark for the web people search task. In: Proc. of the 4th International Workshop on Semantic Evaluations - SemEval 2007, Association for Computational Linguistics (2007) 64–69
2. Artiles, J., Gonzalo, J., Sekine, S.: Weps 2 evaluation campaign: overview of the web people search clustering task. In: Proc. of the 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference. (2009)
3. Stein, B.: Principles of hash-based text retrieval. Clarke, Fuhr, Kando, Kraaij, and de Vries, Eds., 30th Annual Int. ACM SIGIR Conf. (2007) 527–534
4. Shin, K., Han, S.Y.: Fast clustering algorithm for information organization. In: CICLing. Volume 2588 of LNCS., Springer-Verlang (2003) 619–622
5. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval **12**(4) (2009) 461–486

Image Processing and Texture Analysis

Periodicity and Texel Size Detection Using Sum and Difference Histograms

Rocio A. Lizarraga-Morales, Raul E. Sanchez-Yanez, and Victor Ayala-Ramirez

Universidad de Guanajuato DICIS
rocio_lizarraga@laviria.org, {sanchezy, ayalav}@salamanca.ugto.mx
Salamanca, Guanajuato, MEXICO

Abstract. Texture periodicity and texture element (texel) size are important characteristics for texture recognition and discrimination. In this paper, an approach to determine both, texture periodicity and texel size, is proposed. Our method is based on the entropy, a texture measure computed from the Sum and Difference Histograms. The entropy value is sensitive to histograms parameters and takes its lower value when the parameters match with texel size or its integer multiples, in an specific direction. We show the performance of our method by texture synthesis, tiling a sample of the detected size and measuring the similarity between the original image and the synthesized one, showing good results with regular textures and texels with different shapes.

Key words: Texel Size, Texture Periodicity, Entropy, Texture Analysis, Texture Synthesis.

1 Introduction

Texture analysis is an important issue in computer vision. Texture is a visual property perceived in all objects around us. There is not a formal definition of visual texture, but from the structural point of view, it is widely accepted to define it as an image conformed by two components: a texture element (texel), which is the fundamental micro-structure in generic natural images [14], and its placement rules. The main task of structural analysis is to identify texels and their placement rules, which can be related with texture periodicity. These components are sought in order to obtain a compact description of the analyzed texture, which could be used in applications of texture synthesis [2], high speed image transmission, texture compression, among others. In the same way, the texel can be used as reference in order to improve performance in classification [5][8] and segmentation [12] tasks and achieve invariant texture analysis [6]. In the case of natural textures it is not trivial to analyze periodicity since it is reduced by variations in elements, like illumination changes, perspective deformations, shape, and same stochastic nature of texture.

Coocurrence matrix (CM) proposed by Haralick [4] has been widely used to detect periodicity exploiting its parametrization, and mainly using two statistics: χ^2 and κ [11]. Other approaches have been proposed for the same task. Textural

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 89-96

periodicity determination using a distance matching function has been proposed by Oh *et al.* [10], improving time consumption in comparison with inertias from the CM. Recently, Ahuja and Todorovic [1] have proposed the extraction of texels in 2.1D natural textures, using segmentation trees. Grigorescu and Petkov [3], estimate the minimum square window that corresponds with texel size of regular images based on the calculation of Renyi's generalized entropies. Leu [7] has proposed the use of the gradient field and an autocorrelation function to determine the periodicity of a given texture. Nang and Pang [9] have proposed the regularity analysis of texture with a regular bands (RB) method.

In this paper, texture periodicity and texel size are determinated using the property of entropy calculated from the Sum and Difference Histograms (SDH) proposed by Unser [13]. SDH has a free parameter, a relative displacement vector (DV) which consists in two components, horizontal and vertical. When DV is varied, an entropy function with respect to DV is obtained. When DV matches with the texel size or an integer multiple, entropy function reaches local minima, such behavior allows us to detect period and hence, the texel size. Moreover, SDH present computational advantages over the CM in complexity and memory consumption since SDH decrease in one order the CM memory storage. The paper is structured as follows: Section 2 describes the calculation of SDH. Section 3 describes the proposed method to extract the periodicity and texel size. In Section 4 the experimental setup is described and results for artificial and natural textures with different randomness levels are presented in Section 5. Conclusions and future work are given in Section 6.

2 Sum and Difference Histogram

To obtain the SDH, let us define a texture image $I(m,n)$ of $M \times N$ pixels size, and K gray levels $k = \{0, 1, \ldots, K-1\}$. Let us consider a pixel localized in coordinates (m,n) with intensity denoted as $I_{m,n}$ and a second pixel in a relative position with intensity $I_{m+d_m,n+d_n}$, where (d_m, d_n) is the relative displacement vector (DV). Sum and difference, associated with the relative DV (d_m, d_n), are defined as:

$$s_{m,n} = I_{m,n} + I_{m+d_m,n+d_n}, \tag{1}$$

$$d_{m,n} = I_{m,n} - I_{m+d_n,n+d_n}. \tag{2}$$

Sum histogram h_s and difference histogram h_d with displacement vector (d_m, d_n) over the image domain D, are defined as: $h_s(i) = Card\{(m,n) \in D, s_{m,n} = i\}$ and $h_d(j) = Card\{(m,n) \in D, d_{m,n} = j\}$. Normalized SDH are estimations of the sum and difference probability functions $P_s(i)$ and $P_d(j)$.

Different measures computed from the probability distributions have been proposed to be used as textural information. In this work, the measure used is the entropy, defined in Eq. 3.

$$\text{entropy} = -\sum_i P_s(i) \cdot log(P_s(i)) - \sum_j P_d(j) \cdot log(P_d(j)). \quad (3)$$

3 Periodicity and Texel Size Determination

As was described previously, SDH are parameterized by a displacement vector which is represented in cartesian coordinates as (d_m, d_n). Let us take the periodic texture $I(m,n)$ previously described, with dimensions of $M \times N$ pixels and period T_m, T_n in horizontal and vertical directions, respectively. Using SDH we can obtain an entropy value $E(d_m, d_n)$ in function of the displacement vector $DV = d_m, d_n$. In a periodic texture, when DV matches the texture period (texel size), or multiples of it, entropy function reaches a minimum value. This behavior can be exploited in order to detect period for horizontal and vertical directions, when one of the parameters is fixed to zero. When $d_m = 0$, d_n takes values in the range $[2, \frac{N}{2}]$, and when $d_n = 0$, d_m is in the range of $[2, \frac{M}{2}]$. The upper value is proposed to assure at least two texels in the surface analyzed.

For example, an artificial and periodic texture with a square texel of 32×32 pixels size is presented in Fig. 1a. In Fig. 1b the entropy plots in both directions horizontal (with the marker (o)) and vertical (with the marker (+)) are given. We can see in this case, that the entropy function is the same in both directions, since this is a square texel. We also can see, that the entropy plot is periodic and has minima when DV matches with integer multiples of the texel size (e.g., 32, 64, 96,...).

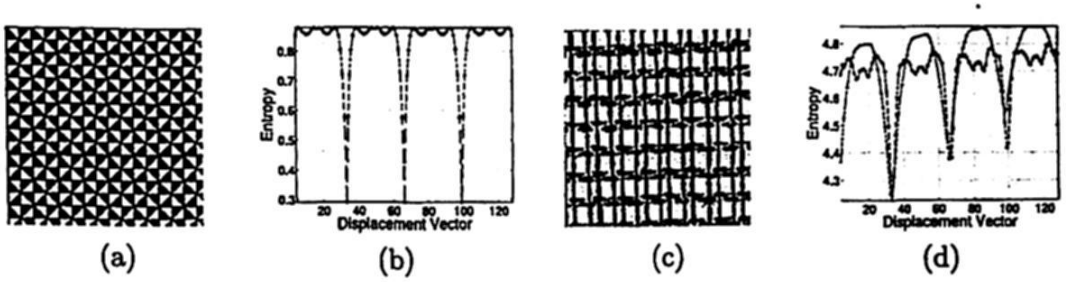


Fig. 1. An artificial texture pattern (a) and its entropy function in both directions (b). A natural texture pattern (c) and its entropy function in both directions (d).

We can determine the texel size of an artificial texture pattern by finding the first global minimum, but it is more difficult when a natural texture is analyzed. Changes in texel shape and its placement rules reduce periodicity due to the typical irregularities said before. Lets take the natural texture pattern shown in Fig. 1c, whom texel size is approximately 33×33 pixels size. By analyzing this

texture pattern with the same criteria, we obtain the entropy functions shown in Fig. 1d. Although functions are not periodic, they still have local minima when DV corresponds with the texel size. Including natural cases in the general criteria, we can say that texel size is determined by the DV where the global minimum from entropy functions is found.

4 Experimental Setup

To test the performance of our approach, we use 9 images: synthetic textures, natural quasi-periodic textures and a natural stochastic texture. The image data set (see Fig. 2) consists of an artificial texture and 8 images extracted from the Brodatz album [?]. Take notice that although natural images are apparently regular, a detailed inspection shows that the position rules vary, considering them as quasi-periodic. Last image (D9), can be defined as stochastic since it looks like 2D noise.

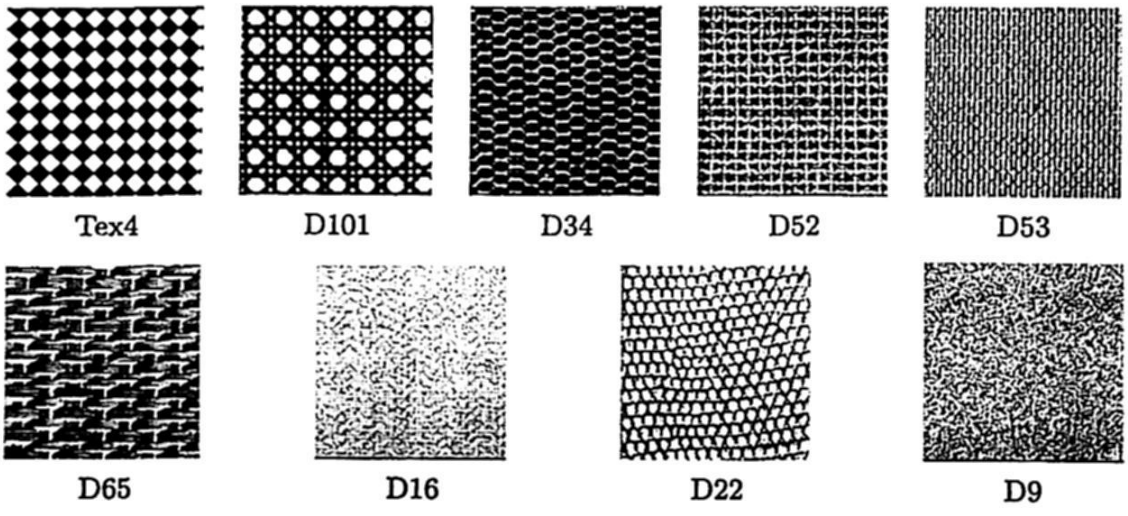


Fig. 2. Data Set used in experiments.

Performance is evaluated by taking a sample from the original image and using it in texture synthesis. Texture synthesis has been an important issue in computer vision, used as a way to verify texture analysis methods among other applications. The algorithm used in texture synthesis is tiling. Such algorithm consists in tiling an image with specific dimensions where the tile is a sample of a given texture. In this experiment, the tile is the sample taken from the original image. In order to quantify similarity between the original image and the synthetic image we use the Kolmogorov-Smirnov (KS) test defined in Eq. 4. KS test has values in the range [0,1] so we can have an intuitive result where zero is the value for two images that match exactly and one for two completely different images, like an image completely white and other completely black.

$$KS = max_k|H_1(k) - H_2(k)|, \tag{4}$$

where H_1 and H_2 are the cumulative distribution functions of two histograms.

5 Experimental Results

Entropy functions were calculated for each image, their plots are presented in Fig. 3, horizontal direction is presented with the mark (o) and vertical direction is presented with the mark (+). Table 1 shows the results of texel size detection, following the criterium that the texel size is found in the DV which corresponds to the global minimum.

In this image set, we can find four different cases. First case is the artificial texture named Tex4. This image is periodic in both directions, so entropy functions are periodic too. Since there are not a single global minimum, texel is detected in the first global minimum. In natural textures, there are specific cases, the first one is a quasi-periodic texture with square texel (e.g., D101). Its entropy functions show some periodicity in both directions and the global minimum in each direction matches with texel size. Another case is when there are periodicities in both directions but not with the same period, this means that the texel has a rectangular shape (e.g., D34, D52, D53, D65, D16). Last case within natural textures, is when there is not apparent regularity or it is difficult to describe a basic pattern and its position rules. This is the case of D9, where entropy plots looks like noise, the global minimums are localized in a high DV, hence the texel size tends to be big.

Table 1. Results in texel size determination and Kolmogorov-Smirnov test of synthesized textures and original textures.

Texture	Texel	KS Test	Texture	Texel	KS Test
Tex4	28 × 28	0.000	D65	28 × 69	0.110
D102	38 × 37	0.027	D16	7 × 43	0.044
D34	17 × 41	0.284	D22	32 × 15	0.110
D052	59 × 32	0.284	D9	83 × 109	0.052
D53	20 × 14	0.247			

In Fig. 4 the synthetic textures created from the texel detected are shown. The synthesis algorithm consists in taking the first sample of the size detected and repeating this sample in both horizontal and vertical direction through a specific surface. Results of KS test are shown in Table 1. These values show that the synthetic image is very similar to the original one with values lower than 0.3. We can see that the synthetic texture of the image named as Tex4, matches exactly with the original one obtaining a value of zero. Synthetic images

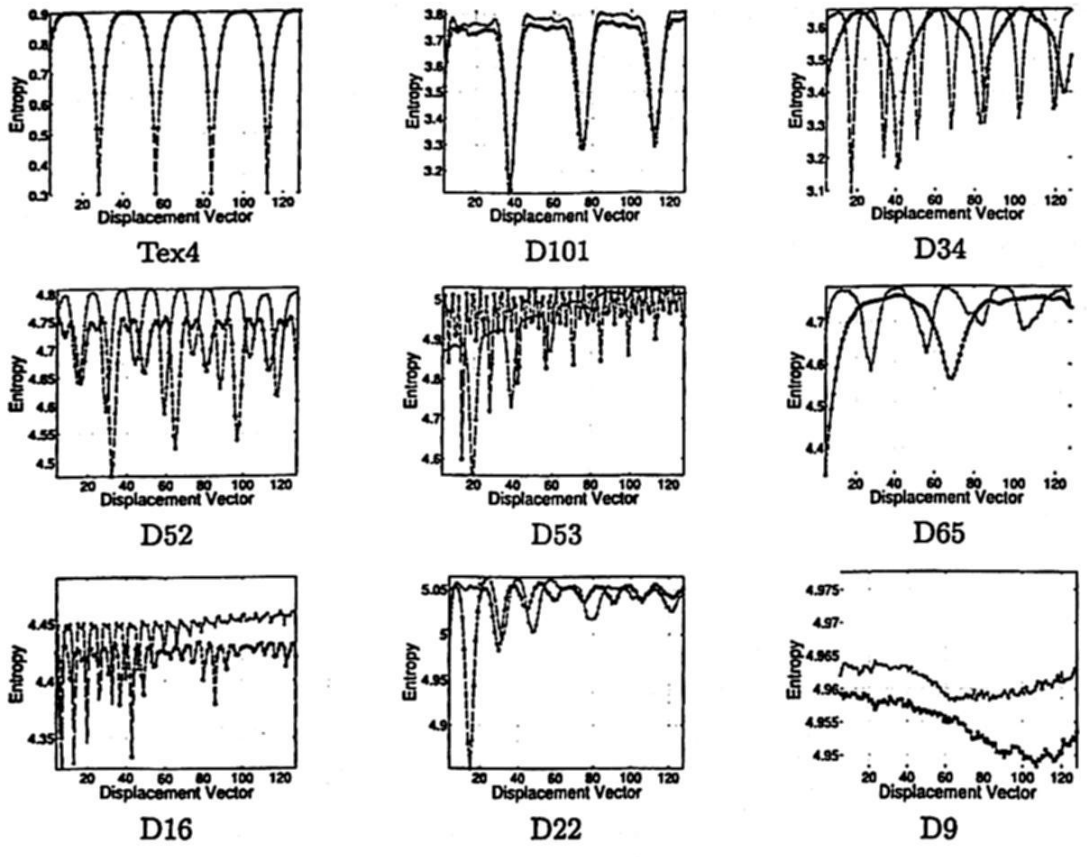


Fig. 3. Entropy plots for each image in horizontal (o) and vertical (+) directions.

corresponding with D101, D052, D16 and D9, exhibit good similarity values, lower than 0.1, this is, images are similar in more that 90%. The highest value of 0.28 is obtained with the image D34. In general, due to natural irregularities of texture, there are certain differences between the original and synthetic images. As tiling algorithm is regular, there is an error that accumulates and propagates through all the synthetic surface.

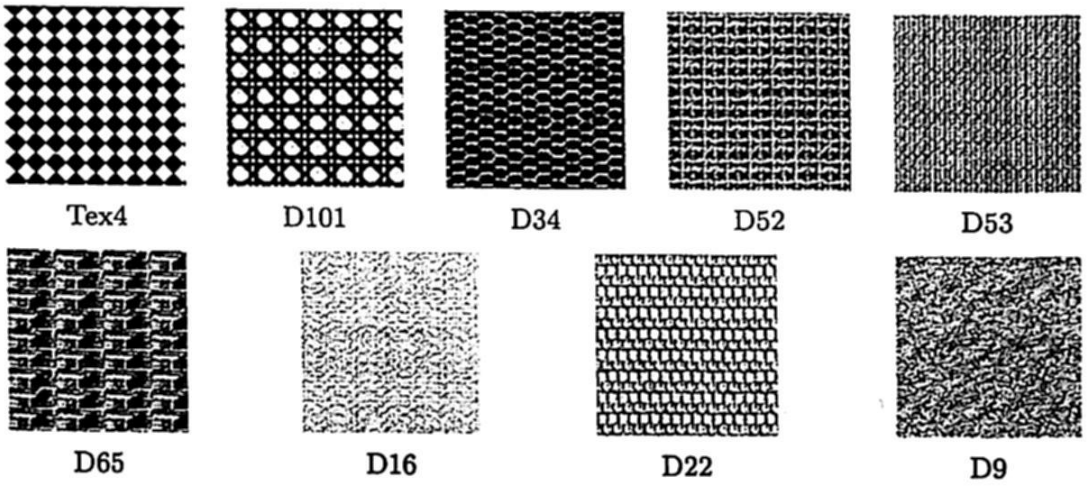


Fig. 4. Synthesized textures by tiling detected texels. Compare with Fig. 2.

6 Conclusion

Periodicity and texel size detection are classic problems in structural texture analysis. In this paper a method based on the property of entropy, calculated from the SDH, to detect the texel size was proposed. Entropy is calculated as a function of the DV in SDH. When DV matches texel size or its multiples, the entropy function has local minima, but texel size is found in the DV where the global minimum is found. Performance was evaluated qualitatively with texture synthesis by tiling, and quantitatively with the Kolmogorov-Smirnov test, showing good results with periodic, quasi-periodic and stochastic textures.

The main advantage of this method over the based on the CM is the use of the SDH that simplifies the computational complexity and memory storage. In an image with K grey levels, the CM requires $K \times K$ memory locations, while the SDH only requires $2K - 1$. Other advantage is the analysis in two directions by fixing to zero a component of the DV. Doing so, we can detect different shapes of texels, accurately identifying a squared or rectangular texel.

References

1. N. Ahuja and S. Todorovic. Extracting texels in 2.1.d natural textures. *11th. IEEE Int. Conf. on Computer Vision*, 2007.
2. Weiming Dong, Ning Zhou, and Jean-Claude Paul. Tile-based interactive texture design. In *Pro. of the 3rd Int. Conf. on Technologies for E-Learning and Digital Entertainment*, pages 675–686, Berlin, Heidelberg, 2008. Springer-Verlag.
3. S.E. Grigorescu and N. Petkov. Texture analysis using Renyi's generalized entropies. In *Proc. IEEE Int. Conf. on Image Processing (ICIP) 2003*, volume 1, pages 241–244. IEEE, 2003.
4. R.M. Haralick. Statistical and structural approaches to texture. *Proc. on the IEEE 4th. Int. Joint Conf. Pattern Recognition*, pages 45–60, 1979.
5. Sen-Ren Jan and Yuang-Cheh Hsueh. Window-size determination for granulometrical structural texture classification. *Pattern Recogn. Lett.*, 19(5-6):439–446, 1998.
6. G.R. Kishor, D.P. Mital, and W.L. Goh. Invariant texture analysis based on attributes of texture elements. In *Proc. of the IEEE Int. Symp. on Industrial Electronics (ISIE '95)*, volume 1, pages 400–404. IEEE, 1995.
7. Jia-Guu Leu. On indexing the periodicity of image textures. *Image and Vision Computing*, 19(13):987–1000, 2001.
8. R.A. Lizarraga-Morales, R.E. Sanchez-Yanez, and V. Ayala-Ramirez. Optimal spatial predicate determination of a local binary pattern. In *Proc. of the 9th Int. Conf. on Visualization, Imaging and Image Processing (VIIP'09)*, pages 41–46. Acta Press, 2009.
9. Henry Y. Y. Ngan and Grantham K.H. Pang. Regularity analysis for patterned texture inspection. *IEEE Trans. on Automation Science and engineering*, 6(1):131–144, 2009.
10. Gyuhwan Oh, Seungyong Lee, and Sung Yong Shin. Fast determination of textural periodicity using distance matching function. *Pattern Recogn. Lett.*, 20(2):191–197, 1999.
11. Valery V. Starovoitov, Sang-Yong Jeong, and Rae-Hong Park. Texture periodicity detection: features, properties, and comparisons. *IEEE Trans. on Systems, Man, and Cybernetics, Part A*, 28(6):839–849, 1998.
12. Sinisa Todorovic and Narendra Ahuja. Texel-based texture segmentation. In *Proc. of the Int. Conf. on Computer Vision (ICCV '09)*, 2009.
13. M. Unser. Sum and difference histograms for texture classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):118–125, 1986.
14. S. Zhu, Cheng en Guo, Yizhou Wang, and Zijian Xu. What are textons? *International Journal of Computer Vision*, pages 121–143, 2005.

Person Verification Process Using Iris Information

Mireya S. García-Vázquez[1], Alejandro A. Ramírez-Acosta[1]

[1] Instituto Politécnico Nacional-CITEDI, Av. Del Parque No.1310, Tijuana BC,
mgarciav@citedi.mx, alvaro@citedi.mx

Abstract. Biometric verification systems employing images of the iris are claimed to be extremely accurate. The iris biometrics features can be used on security systems applications, due to their great advantages, such as variability, stability and security. The speed and performance of an iris verification system is crucial. Thus in this paper, efficient verification process for iris recognition with important performance in feature extraction stage and high confidence is described. The proposed system uses 1D Log-Gabor filter for texture analysis. Experimental tests were performed using CASIA iris database.

Keywords: Iris, Recognition, biometrics.

1 Introduction

Iris recognition is one of important biometric recognition approach in a human identification. Using only iris pattern information, it has a higher accuracy rate than other biometric recognition methods such as face recognition, fingerprint recognition, voice recognition, and hand geometry, noting that an iris has much pattern information and is invariable through a lifetime [1-4]. Thus, iris recognition is in the limelight of security applications. There are mainly two iris recognition prototypes that had been developed: one by J. Daugman [1-2], and the other by W. Wildes [3]. Most of the commercial iris recognition systems are based on these prototypes [4-7]. However, one of the most successful commercial iris recognition systems is distributed by L1 identity solutions (known before as Iridian Technologies) [5] and it uses algorithms patented by Daugman.
Eye images are captured by the system (Image Acquisition). Then, they are processed by first detecting and segmenting the iris (Preprocessing). The iris region of the image is warped (unrolled) to normalized rectangle. Features that represent the iris patterns are extracted and a code (template) is generated (Encode Features). And finally, verification is then achieved by matching the template against a gallery of templates using normalized Hamming distance (Matching codes).
The objective of this article is to describe an efficient verification process for iris recognition with important performance in feature extraction stage and high confidence. In this article the different concepts for every stage in an iris recognition system were

© J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad and H. Sossa (Eds.)
Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 97-104

developed in C language in order to verify the system performance and to validate the chosen options for every stage. The Section 2 describes and analyzes each stage of the system that has been implemented. The experiments and results will be presented in the section 3. Finally conclusions will be drawn in section 4.

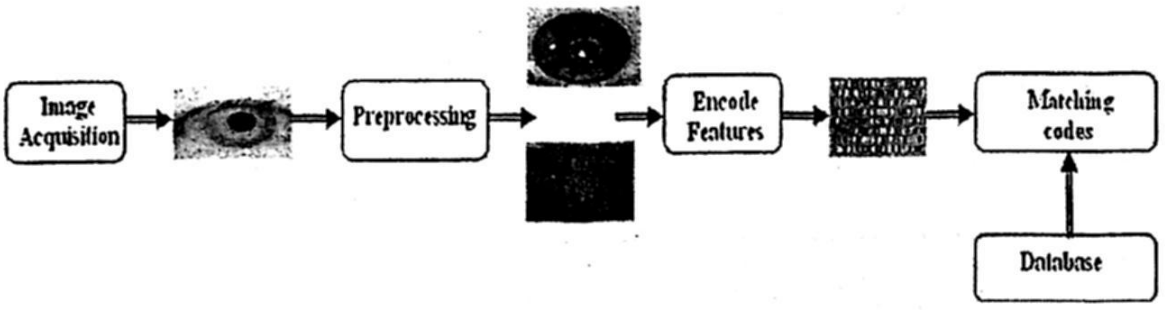


Fig. 1. How iris verification process work.

2 Iris Verification Process

2.1 Image Acquisition

One of the major challenges of automated iris verification system is to capture a high quality image of the iris while remaining noninvasive to the human operator. Given that the iris is a relatively small (1cm in diameter), dark object and that the people are very sensitive about their eyes, this matter required careful engineering. The following points should be concern: Desirable to acquire images of the iris with sufficient resolution and sharpness to support verification. It is important to have good contrast in the interior iris pattern without resorting to a level of illumination that annoys the subject. The images should be well framed (i.e. centered). Noises in the acquired images should be eliminated as much as possible.

2.2 Iris Localization

An eye image contains not only the iris region but also some unuseful parts, such as the pupil, eyelids, sclera, and so on. For this reason, at first step, segmentation will be done to localize and extract the iris region from the eye image. Iris localization is the detection of the iris area between pupil and sclera (see Fig. 2.a). So we need to detect the upper and lower boundaries of the iris and determine its inner and outer circles. The captured image is a 2-D array (M x N) and is described as I(x, y) where the point (x, y) is the gray-level. The first eye part to isolate is the pupil (dark circular area in an eye image, see Fig 2.b). It is used to fix a region threshold to binarize the image I(x,y) to obtain an image $I_B(x, y)$. The image is scanned in the horizontal direction to find the longest chord which will be the diameter of the pupil. The following equations are used to find the centre coordinates [8]:

$$x_p = x + \frac{Dia(x,y)}{2} \qquad (1)$$
$$y_p = y$$

Dia is the longest chord, and (x, y) is the start point of Dia.

The outer boundary of the iris is more difficult to detect because of the low contrast between the two sides of the boundary. Canny edge detection is performed to create an edge map to generate gradients information [9]. Circular Hough Transform which is employed by Wildes [10], is used to detect the iris-sclera boundary to obtain a new centre (x_s,y_s) and radio r_s .The linear Hough Transform [11] is used to detect and to isolate eyelids and eyelashes.

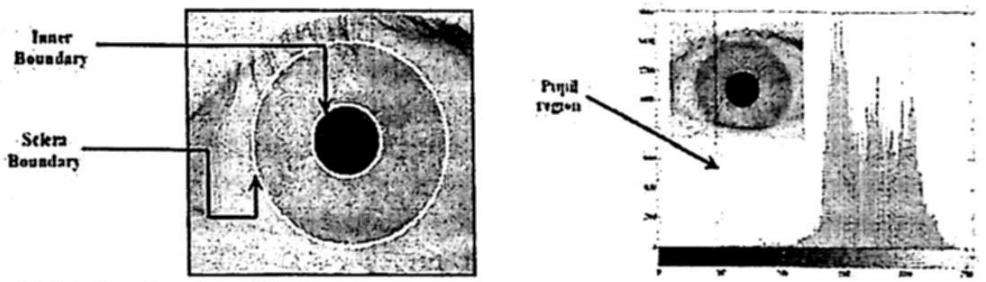


Fig. 2. (a) Iris localization. (b) Gray levels for the pupil region.

2.3 Iris Normalization

The irises captured from the different people have different sizes. The size of the irises from the same eye may change due to illumination variations, distance from the camera, or other factors. At the same time, the iris and the pupil are non concentric. These factors may affect the result of iris matching. In order to avoid these factors and achieve more accurate verification, the normalization for iris images is implemented. In normalizations, the iris circular region is transformed to a rectangular region with a fixed size. With the boundaries detected, the iris region is normalized from Cartesian coordinates to polar representation. This operation is done using the following operation:

$$I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta) \qquad (2)$$

with

$$x(r,\theta) = (1-r)x_p(\theta) + rx_s(\theta) \quad \textit{and} \quad y(r,\theta) = (1-r)y_p(\theta) + ry_s(\theta)$$

The r is on the interval [0,1] and θ is an angle [0,2π]. The I(x,y) is the region image, (x,y) are the original Cartesian coordinates, (r,θ) are the corresponding normalized polar coordinates. The x_p,y_p and x_s,y_s are the coordinates of the iris ring along the θ direction.

Besides to generate the normalized iris image (iris template) is necessary to generate another image called "noise mask". This mask indicates the regions of the normalized iris where the pattern of the iris is obstructed by the eyelids, eyelashes, etc. The noise mask has the same dimensions that the iris image, it is used in the comparison stage to avoid to compares the obstructed regions.

2.4 Feature encoding

The iris has a particularly interesting structure and provides abundant texture information. So, it is desirable to explore representation methods which can describe global and local information in an iris. In this stage the iris features are obtained convolving the normalized iris pattern with a 1D Log-Gabor filters. Gabor Filters based methods have been widely used as feature extractor in computer vision, especially for texture analysis [12]. Gabor filters can serve as excellent band-pass filter for one-dimensional signals. Daugman [1,15] used multi-scale Gabor wavelets to extract phase structure information of the iris texture. However, Field [14] has examined that there is a disadvantage of the Gabor Filter in which the even symmetric filter will have a DC component whenever the bandwidth is larger than one octave. To overcome this disadvantage, another type of filter is used known as Log-Gabor, which is Gaussian on a logarithmic scale, can be used to produce zero DC components for any bandwidth. The Log-Gabor filters are obtained by multiplying the radial and angular components together where each even and odd symmetric pair of Log-Gabor filters comprises a complex Log-Gabor filter at one scale. The frequency response of a Log-Gabor Filters is given as:

$$G(f) = e^{\left(\frac{-(\log(f/f_0))^2}{2(\log(\beta/f_0))^2}\right)} \tag{3}$$

Where f_0 represents the central frequency, and β bandwidth of the filters.

The 1D Log-Gabor Filter is chosen to be the feature extractor of iris for the implementation since 1D Log-Gabor Filters is an improved version of Gabor Filters. By applying 1D Log-Gabor Filters, 2D normalized pattern is divided into a number of 1D signals, and these 1D signals are convolved with 1D Gabor wavelet. The rows of the 2D normalized pattern are taken as the 1D signal; each row corresponds to a circular ring on the iris region. The angular direction is taken rather than the radial one, which corresponds to columns of the normalized pattern, since maximum independence occurs in the angular direction. To encode each row of the image (X vector), the discrete Fourier transform (DFT) is applied to get vector Y (eq.4). Then vector Y and a 1-D log-Gabor wavelet are multiplied to get the vector Z (eq.5). The 1D Log-Gabor function is defined (eq.3), only one filter is used, with f_0 =12, and a bandwidth $\beta = 0.5$ [15]. Finally, using the inverse DFT on the vector Z to get the vector D (eq.6).

$$Y_k = \sum_{n=1}^{N} x_n e^{-j2\pi(k-1)(n-1)/N} \tag{4}$$

$$z_k = Y_k \times G_k \tag{5}$$

$$D_k = \frac{1}{N}\sum_{n=1}^{N} z_k \, e^{-j2\pi(k-1)(n-1)/N} \tag{6}$$

The output of the Gabor filter are complex numbers, where the phase angle at each output point is quantized to two bits depending on the quadrant where this each element in the complex plane. Thus for an image normalized with size (MxN) the resulting template is of size (Mx2N). The template size with radial resolution of 128 pixels and angular resolution of 256 pixels was chosen. These parameters generate an iris template that contains 32768 bits of information.

2.5 Matching codes

Matching is a process to determine whether two iris templates are from the same individual. To perform the matching process, we will attempt to measure the Hamming Distance (HD) [1] between two iris templates. The Hamming distance between any two equal length binary vectors is simply the number of bits positions in which they differ divided by the length of the vectors.

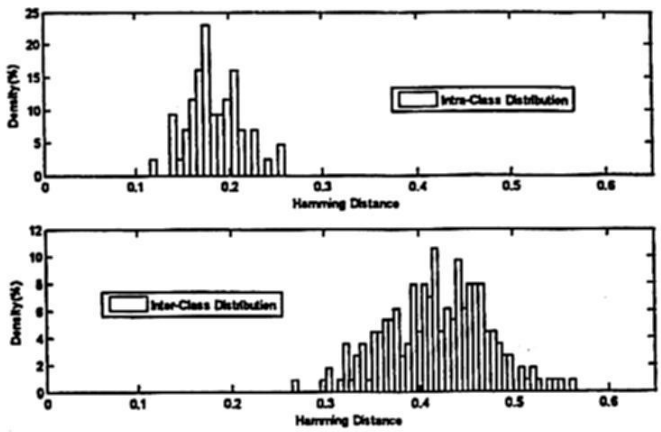


Fig. 3. Distributions of probability intra-class and inter-class in function to hamming distance.

Therefore, using the Hamming distance of two bit patterns, a decision can be made as to whether the two patterns were generated from different irises or from the same one. Since an individual iris region contains many features, each iris region will produce a bit-pattern which is independent to that produced by another iris, and two iris codes produced from the same iris will be highly correlated (intra-class comparisons). If two bits patterns are completely independent, such as iris templates generated from different irises (inter-class comparisons), the Hamming distance between the two patterns should equal 0.5. This occurs because independence implies the two bit patterns will be totally random, so there is 0.5 chance of setting any bit to 1, and vice versa. Therefore, half of the bits will agree and half will disagree between the two patterns. If two patterns are derived from the same

iris, the Hamming distance between them will be close to 0.0, since they are highly correlated and the bits should agree between the two iris codes. Although, in theory, two iris templates generated from the same iris will have a Hamming distance of 0.0, in practice this will not occur. Normalization is not perfect, and also there will be some noise that goes undetected, so some variation will be present when comparing two intra-class iris templates.The Hamming distance algorithm employed incorporates noise masking ($C(i,j)$), so that only significant bits are used in calculating the Hamming distance between two iris templates (A,B). Both iris mask (iris *maskA*, iris *maskB)* are used in the calculation. The Hamming distance will be calculated using only the bits generated from the true iris region given by the equation:

$$HD(A,B)=\frac{1}{\sum_{i,j}C(i,j)}\sum_{i,j}\left(A(i,j)\ xor\ B(i,j)\ \&\ C(i,j)\right)$$

where (7)

$$C(i,j)=\begin{cases}1\ if\ maskA(i,j)=0\ \ and\ \ maskB(i,j)=0;\\ 0\qquad\qquad\qquad\qquad\qquad otherwise\end{cases}$$

To make a recognition decision, a threshold X can be defined. If the obtained distance of Hamming is minor who thresholds X, is decided that the compared codes were generated by same eye, otherwise is decided that they were generated by different eyes. Both distributions Inter-class and intra-class, generally, are overlapped. For that reason, the area under the distribution Inter-class to the left of threshold X, represents the probability of a false identification, whereas the area under the distribution intra-class to the right of threshold X, represents the probability of a false rejection.

These measures are known as false acceptance rate (FAR) and false rejection rate (FRR), the FAR is the probability that an unauthorized subject is incorrectly accepted by the system whereas the FRR is the probability that an authorized subject is incorrectly rejected. The distance for both distributions ("decidability") can be calculate by equation (8), this value can be used to optimize the parameters on log-Gabor filters

$$d=\frac{\left|\mu_{INTRA}-\mu_{INTER}\right|}{\sqrt{\sigma_{INTRA}^{\ 2}+\sigma_{INTER}^{\ 2}/2}} \tag{8}$$

where

μ_{INTRA}, μ_{INTER} = the mean of the probability distributions

$\sigma_{INTRA}, \sigma_{INTER}$ = standard deviation of the probability distributions

3 Experiments

In order to evaluate the iris verification algorithms, CASIA (The Chinese Academy of Sciences Institute of Automation) iris image database is used [16]. This image database contains 756 eye images from 108 different persons. The experiments were performed in

C language on Pentium IV PC with 512 MB RAM and 1.8 GHz. The experiments were obtained with verifications, evaluating genuine matching scores for all the possible combinations (60 intra-class comparisons) and evaluating the impostor matching scores for all the possible combinations (190 inter-class comparisons). The parameters distributions obtained were $\mu_{INTRA} = .182$ $\sigma_{INTRA} = .029$ for Intra-class distribution and $\mu_{INTER} = .4195$ $\sigma_{INTER} = .0534$ for Inter-class distribution. The calculated distance between the intra-class and the inter-class distribution was d=5.52732, and the portion that overlaps between the intra-class and the inter-class was very small (see fig. 3). This proves that the proposed features are highly discriminating. Acceptance Rate (FAR) and False Rejection Rate (FRR) are the two critical measurements of system effectiveness. The values of FAR and FRR were 0.182516% and 0.185026% respectively. The EER (Equal Error Rate) was 0.202 %.The iris features (320 entries) were extracted with a bandwidth of $\beta = 0.5$ with center wavelength f_0=14 pixels.

The method described in this implementation locates the iris in the image based on edge detection approach, whereas Daugman [1] makes use of a deformable model. In addition, several tests were made to obtain the time of execution for each stage in the implemented system. The average total execution time of a basic verification process not exceeds 500 ms, which is suitable for a recognition system.

The table 1 shows the comparative results of the time computation consumed by each block of the iris verification algorithms of four methods. As we can see, the experimental results have shown that the computational cost for Extraction features stage of our method consumes less time than the other methods (12.6ms). The Log-Gabor filter produces an uncorrelated and less redundant representation for iris texture compared with the ordinary Gabor Filters. Now, we are working on more precisely in Iris localization optimization (362.8 ms). Thus, we expect to further improve the performance of the current method.

Table 1. Time consuming for each stage of the different implemented systems

Operation	Daugman [2]	Tisee et all [17]	Avila et all [12]	Wildes [10]	Proposed
Iris localization	90 ms	250 ms	—	10 seg	362.8ms
Extraction features	102 ms	188 ms	229.5 ms		12.6 ms

4 Conclusions and Perspectives

The work presented in this paper involves the implementation of all phases of an iris recognition system. In order to verify the system performance and to validate the chosen options for every stage, the iris recognition system was developed in C language. The implemented algorithm uses iris information from both global and local iris data base. Each iris image is filtered with Log-Gabor filter and then a fixed length feature vector is obtained. The evaluation of the system achieved high confidence identity verification

based on iris texture using Log-Gabor transform; further efforts should be applied to improve iris localization stage.

Acknowledgment

This work was support by IPN- SIP20090041.

References

1. J. G. Duagman. "High confidence visual recognition of person by a test of statistical independence". IEEE Trans. Pattern Analysis and Machine Intelligence. Vol.25. pp. 1148–1161. Nov. 1993.
2. Daugman J. "New methods in iris recognitions". IEEE Transactions on Systems. Man and Cybernetics –B 37. pp. 1167-1175. Oct. 2007.
3. R. P. Wildes. "Iris recognition: an emerging biometric technology". Proc. IEEE. Vol.85. pp. 1348–1363. Sep. 1997.
4. L. Ma, T. Tan, Y. Wang, and D. Zhang. "Personal identification based on iris texture analysis". IEEE Trans. Pattern Analysis and Machine Intelligence. Vol.85. pp. 1519–1533. Sep. 1997.
5. L1 identity solutions. Available from: http://www.l1id.com/pages/17, 2009 (accessed March 2009).
6. LG. Available form: http://www.lgiris.com/, 2009 (accessed March 2009).
7. Sagem Morpho. Available from: http://www.morpho.com/, 2009 (accessed March 2009).
8. Li Yu, David Zhang, Kuanquan W. "The relative distance of key point based iris recognition". Pattern Recognition. Vol.40. pp. 423-430. 2007.
9. J.F.Canny. "Finding edges and lines in images". M.S. thesis. Mass. Inst. Technologies. 1983.
10. R.P.Wildes, J.C.Asmuth, G.L.Green. "A System for Automated Recognition". 0-8186-6410-X/94, IEEE. 1994.
11. P.V.C.Hough. "Method and means for recognizing complex patterns". U.S. Patent 3 069 654, 1962.
12. Sánchez Avila, C Sánchez Reíllo. "Two Different Approaches for Iris Recognition using Gabor Filters and Multiscale Zero-Crossing Representación". Pattern Recognition. Vol.38.n°23. July 2004.
13. John Daugman, "Uncertainly relation for resolution in space, spatial frequency, and orientation optimized by two dimensional visual cortical filters". J Opt. Soc. Amer. A. Vol.2. pp.1160-1169. 1985.
14. D. Field. "Relations between the statics of natural images and the response propierties of cortical cells". Journal of the Optical Society of America. 1987.
15. Chong S, Andrew T, David N. "High security Iris verification system based on random secret integration". Computer Vision and Image Understanding. Vol 102, Issue 2. May 2006. pp.169-177.
16. "CASIA" iris image database collected by institute of automation, Chinese academy of sciences. Available: www.sinobiometrics.com.
17. Tisee et all. "Person identification technique using human iris recognition", Proc of Vision Inteface,pp.294-299, 2002.

Methodologies to Validate Both, the Photo Printing Type and its Spatial Resolution in Personal ID Documents

Samuel Sánchez-Islas[1], Edgardo M. Felipe-Riverón[1], Salvador Godoy-Calderón[2],

[1]Pattern Recognition Laboratory; [2]Artificial Intelligence Laboratory
Center for Computing Research, National Polytechnic Institute
Juan de Dios Batiz w/n and Miguel Othon de Mendizabal, P. O. 07738, Gustavo A Madero, Mexico
(55)57296000 Ext. [1]56557, [2]56515
[1] ssislas@yahoo.com.mx, edgardo@cic.ipn.mx, sgodoyc@cic.ipn.mx

Abstract. This paper introduces two new image-processing methods, the first one determines if the picture appearing on a Personal ID -card is printed in analogical or digital format. The second method determines if the digital photo has a spatial resolution of 800 dots per inch (dpi) . Both methods are tested with the IFE Voting ID-card, which is a "non-official" personal document used as the main identification card in Mexico. Both methodologies were developed by using morphological techniques to extract the useful characteristics by means of a previously defined structure element according to the size on the photo in the document. We tabulated the efficiency of these two methodologies after its use in 30 different ID-cards, 10 out of them analogical and the remainder 20 having digital photograph. All they were genuine ID-cards.

Keywords: Image processing; Photo printing-type; Spatial Resolution; Personal ID documents.

1 Introduction

The Voter Registration Card with photograph is a document issued by the Federal Electoral Institute (IFE-Instituto Federal Electoral), an autonomous agency in Mexico. The Voter Registration Card is used to vote in Federal and State elections in Mexico. In spite of the fact that this is a non-official document, it has become indispensable for Mexicans because it is the only valid ID document issued to adult persons legally that enables them to make bank and money transactions or get access to night clubs and buy cigarettes and alcoholic beverages in the whole country.

There are several locks or security means in the Voter Registration Card with photograph (the "Card") that have been introduced in its design in order to make difficult its forgery [2]. The photograph printed on the Card is an object which has some features incorporated by the IFE to make its forgery almost impossible.

In the Card the photograph may be printed in one of the following two ways:

© J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad and H. Sossa (Eds.)
Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 105-113

analogically, with the photo obtained with a conventional Polaroid camera [3] and digitally, as an image with an 800 dots per inch resolution. In order to confirm the authenticity of the Card, a method to validate the kind of print (analog or digital) and the spatial resolution used to print it in case it was digital, is proposed. The fact that the photo was printed in analogical or digital way depends on the date the ID-card was emitted.

In a state-of-the-art review, it was found that to this date, there is no method to detect the kind of print and to determine the spatial resolution of a digital image.

An analog image can be mathematically represented as a continuous range of values indicating the position and intensity of different objects. The intensity can be represented on a normalized scale from 0 to 1 with infinite divisions, and spatially with an infinite number of two- dimensional coordinates. On the other hand, a digital image is restricted in both its spatial coordinates and in its allowed intensities. Their positions and intensities are represented by finite positive discrete values. The elements that make up a digital image are called picture elements, or pixels.

2 Spatial Resolution

Intuitively, spatial resolution is a measure of the smallest discernible detail in an image. Quantitatively, spatial resolution can be stated in a number of ways, with line pairs per unit distance, and dots (pixels) per unit distance, these being among the most common measures. Suppose that we make a chart with alternating black and white lines, each of W width units (W can be less than 1). The width of a line pair is thus $2W$, and there are $1/2W$ line pairs per distance unit. For example, if the width of a line is 0.1mm, there are 5 line pairs per distance unit (mm). A widely used definition of image resolution is "the largest number of discernible line pairs per distance unit "(e. g., 100 line pairs per mm). "Dots per distance unit" is a measure of image resolution used commonly in the printing and publishing industry. In the U. S., this measure is usually expressed as dot per inch (dpi). To give an idea of the difference in quality, newspapers are printed with a 75 dpi resolution, magazines with 133 dpi, glossy brochures with 175 dpi, and some books with 2400 dpi.

3 Some Basic Morphological Concepts

One of the main applications of the morphological transformations is for removing objects from an image by convolving a preselected structuring element having a given size, shape and orientation and the entire image [7].

Particularly, the top-hat transformation of a gray-scale image f is defined, as the original image f minus its opening (Eq. 1). This transformation allows us to emphasize the clearest zones of the image, so we can get the points where the ink has been printed.

$$T_{hat}(f) = f - (f \circ b) \quad (1)$$

Where $(f \circ b)$ is called the morphological opening.

The opening of a set A by a structuring element B is denoted $A \circ B$ and defined as indicated in Eq.(2):

$$f \circ b = (f \ominus b) \oplus b \quad (2)$$

Thus, the opening A by B is the erosion of A by B, followed by the dilation of the result by the same structuring element B. The erosion is defined as indicated in Eq. (3):

$$[f \ominus b](x,y) = min_{(s,t)\in b}\{f(x+s,y+t)\} \quad (3)$$

On the other hand, the dilation is defined as, Eq. (4):

$$[f \oplus b](x,y) = max_{(s,t)\in b}\{f(x-s,y-t)\} \quad (4)$$

4 Methodologies

During the analysis of the printed photograph it is necessary to take into account the following::

- The kind of print (analog or digital) (Figure 1);
- If the image is digital, the spatial resolution with which it was printed.

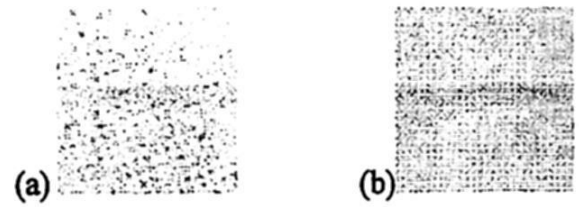


Fig. 1. Kinds of image: (a) digital (b) analog.

In order to find out the kind of print, basic morphological operators were applied to the digital images represented by the photo in the Card. The following flowchart shows the procedure to find out whether the image printed on the Card is analog or digital (Fig. 2).

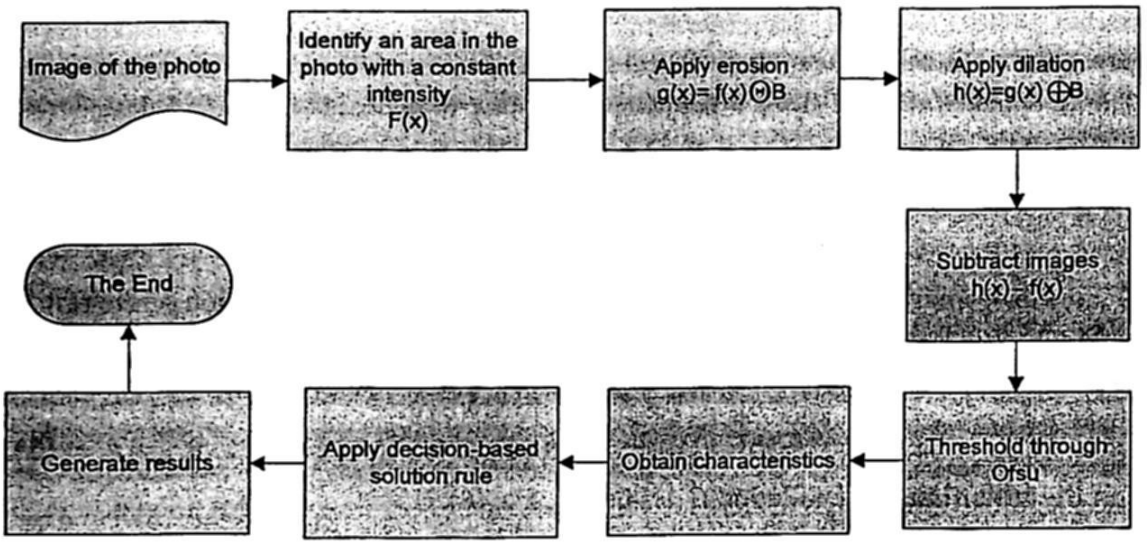


Fig. 2. Flowchart to find out whether the image is analog or digital.

Firstly, a small area with a constant intensity of pale shades is identified in the scanned photo located in the citizen's Card. In our case, 75 x 75 pixels square area was used, equivalent to a 1/16 inch square area in line with the given spatial resolution with which the image was scanned (1200 dpi). An erosion with a 4 x 4 pixels square structure element is applied to this small square area with a reference point at the center; then the resulting image is dilated with the same structure element; finally the resulting image is subtracted from the original image and thresholded by the Otsu method [6]. Once the final image is obtained, the following features are checked:

- If the dots are homogeneous in size;
- If the printed dots appear to be continuous.

Checking that the dots are homogeneous in size is achieved by enclosing a part of the image within an imaginary circle smaller in size than the original image. Afterward, dilation with a 3 x 3 pixels square structural element with a reference point at the center follows. The counting of black dots is started; if the amount of dots is smaller than a pre-defined threshold, it may be stated that it is a digital photograph because the image shows disjoint dots almost of the same size. This is due to the fact that in a digital image, ink is organized in a more or less constant area dot arrangement. On the other hand, if the resulting image shows dots with different sizes sometimes connected among them, we can be sure that it is an analogical print, because it takes place through chemical processes which prevent the formation of isolated dots.

An analog image appears in the Card as "printed" in a continuous way. In order to check that the dots are continuously arranged and disjoint, the former image is reversed and the connected components are labeled. If the number of colored label (objects) is few,

this fact, together with the feature mentioned above, determines that the image is a digital one.

Once the kind of image has been determined, if the image happened to be digital, the next thing to do is to determine the spatial resolution with which it was printed. This should be 800 dots per inch, as per the aforementioned security measure. Calculating the spatial resolution is extremely important, because the Federal Electoral Institute imposed this value due to the fact that having commercial printers to print images with this spatial resolution would not be possible. The flowchart of the whole process to find out the spatial resolution of the printed digital image is shown below (Fig. 3).

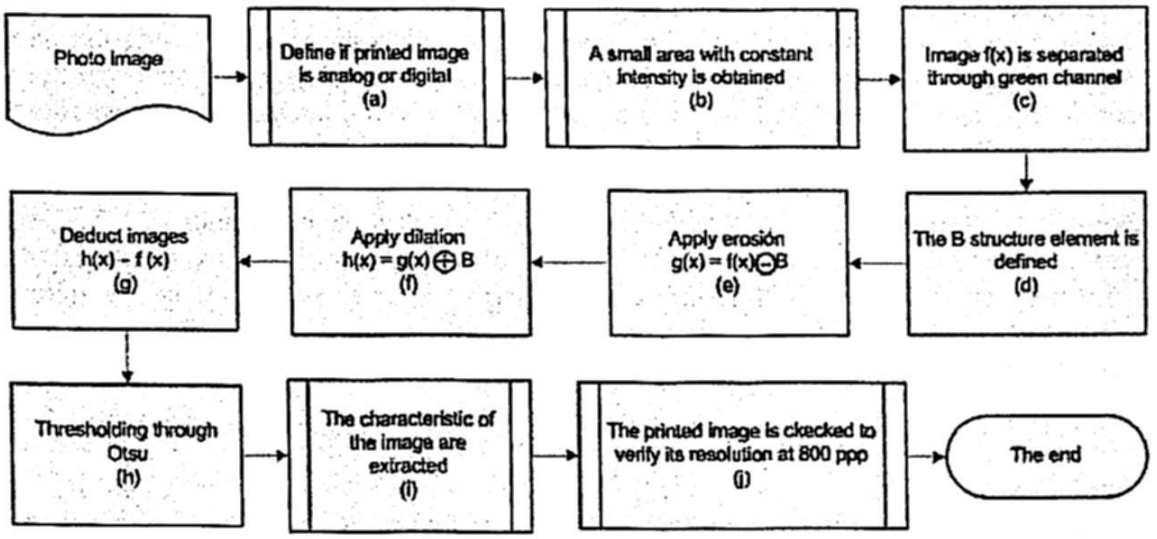


Fig. 3. Flowchart to find out the spatial resolution of the printed photograph.

An explanation of the whole process follows: Once the photograph is obtained, it is determined whether the printed image is analog or digital by using the methodology explained above, (a). If the image is found to be digital, a small square window is cut out from a pale area with more or less constant intensity, (b) with the size found through the Eq. (5).

$$Window\ size = \frac{1}{16} Scanned\ Resolution \tag{5}$$

where the window size is the area in pixels to be analyzed within the image.

The term 1/16 corresponds to the sixteenth part of an inch, which is the length of the side that corresponds to the square area to be analyzed; the scanned resolution is the value in pixels of the resolution with which the image to be analyzed was scanned. In our case, the images were scanned at 1200 dots per inch spatial resolution, it is important to know a priori the scanned resolution from the image in order to make the correct experimentation. If the scanned resolution is unknown it is necessary to calculate it by using specific algorithms or methodologies.

Therefore:

$$Window\ Size = \frac{1}{16}(1200) \tag{6}$$

So:

$$Window\ Size = 75\ Pixels$$

After cutting the entire image the green channel of the color image is selected (c). Once this is done, a cross-shaped structure element (d), whose size depends on the size of the printing dot, is defined. The higher the spatial resolution of the printed photograph the smaller each dot area of the printed image.

Then erosion is applied to the image (e) followed by dilation (f) with the same structure element (Opening). After this step, the original image is extracted from the resulting image (Top-hat), allowing us to see where there is more density of printed dots (g). In the next step the image is thresholded with the Otsu (h) method and the objects are labeled to be able to obtain the characteristics of the resulting image (i); the number of (colored) objects included in the labeled image is counted through Equation (7). These colors are found experimentally with this empiric equation, with which we can get the total number of objects avoiding the colors having similar values when they are counted. The formula calculates different values without repetition, because the range of values is wider than the calculated RGB average.

$$Color = 64(R + 25) + 16(G + 25) + 2(B + 25) \tag{7}$$

Finally, empirical Equation (8) is used to find out if the resolution corresponds to the 800 dots per inch expected in the selected image fragment, in order to be able to assert that the whole photograph is genuine (j).

$$Resolution = \sqrt[2]{Points} * \frac{16}{1} * 2 * \frac{Scanned\ Resolution}{Printed\ Resolution} \tag{8}$$

Where the *Number of Dots* is given by the number of dots counted in the image. The term 1/16 corresponds to one sixteenth of an inch, which is the size of the sides of the square area being analyzed; 2 corresponds to the two dots, a white dot and a black one, which unavoidably make-up the spatial resolution unit. The Scan Resolution refers to the resolution with which the image was scanned and the Print Resolution is the resolution with which the original digital image is supposed to have been printed.

In this case, the image was scanned with a 1200ppp resolution and the printed image must be at 800 dpi in line according to the security measure with which the photograph must have been printed to be considered as genuine. Therefore, the formula would be now as follows:

$$Resolution = \sqrt[2]{Points} * \frac{16}{1} * 2 * \frac{3}{2} \tag{9}$$

This becomes:

$$Resolution = \sqrt[3]{Points} * 48 \quad \textbf{(10)}$$

Once the number of dots in the labeled binary image (previously thresholded by the Otsu algorithm) has been calculated, the final spatial resolution that checks if the photo in the Card was printed at 800 dots per inch is obtained through Eq. (10).

5 Experiments and Results

In order to confirm the proposed method, several experiments with different Card models were carried out. For this purpose, 30 Card images were used: 10 with analog photographs and 20 with digital photographs printed with an 800 dpi resolution. All of them were scanned at 1200 pixels per inch (ppi) by using an HP Scanjet 367Q scanner.

When we applied the first method to find out the kind of image printed on the Card, the results shown in Table 1 were obtained. The results indicate that the method produces a correct classification always in both kinds of prints.

Table 1. Classification results with the first methodology.

Printing type		
Samples	Analog	Digital
30	10	20

When we applied the second method, once we found that the photo printed on the Card was digital, we had to find out if the space resolution of the print was 800 dpi. The results shown in Table 2, show that there is a false negative, because the expected resolution was not obtained due to the fact that when we made the cut, the image with a dark shade was used.

Table 2. Classification results with the second methodology.

	Resolution (dpi)	Number of objects
	800	19
	700	1
Total objects		20

In order to determine the efficiency and yield of both methodologies, measurements were taken with the Precision and Recall methods, having obtained the results included in Table 3, where we may see that in the first method a 100% efficiency is achieved for a

correct classification of each of the analyzed objects. In the second method a 96.6% efficiency is achieved.

Table 3. Efficiency of both developed methodologies.

	Samples (Total)	Correct Classification	Wrong Classification	Coverage	Recall	Precision
Methodology 1	30	30	0	100%	100%	100%
Methodology 2	30	29	1	100%	96.6%	96.6%

6 Conclusion

The methodologies proposed herein allow the reliable and robust determination of the kind of print of the photos included in the Mexican ID-Cards, and in case the print is digital they allow verification of the fact that the spatial resolution corresponds to the one established by the Federal Electoral Institute: 800 dpi. These methodologies based on the top-hat morphological transformation, allows us to emphasize the interesting areas of the image (printed points). The size of the window selected for analysis is empirically defined through a formula. The size of the structure element plays a role in both methodologies, because the higher the expected space resolution, the smaller the structure element required. On the other hand the lower the resolution the larger the size of the structure element. With the first method the efficiency was of 100% and with the second method an efficiency of 96.6% was achieved. These results can be used, together the alphanumeric data printed in the document and related to its proprietary, to decide if an ID-Card is genuine or apocrypha.

Acknowledgements. The authors would like to thank COFAA, Postgraduate and Research Secretary, and Center for Computing Research of the National Polytechnic Institute, CONACyT and SNI, for their economic support to carry out this work.

References

1. Hasta en doce ocasiones han repuesto la credencial del IFE, en: http://noticias.vanguardia.com.mx/d_i_308095_t_Hasta-en-12-ocasiones-han-repuesto-la-credencial-del-IFE.htm
2. Elementos y características de la credencial para votar con fotografía, en http://www.ife.org.mx/documentos/DERFE/RFE2/cred/CredencialVotar_anverso.swf.
3. Diario Oficial de la Federación, México, 30 de septiembre de 1992 (1992).
4. Diario Oficial de la Federación, México, 31 de enero de 2001 (2001).

5. Código Penal Federal, México (2000).
6. N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man and Cybernetics, 9(1):62-66 (1979).
7. R. C. Gonzalez y R. E. Woods, Digital Image Processing, 3rd Edition, Prentice Hall (2008).
8. Sossa Azuela Juan Humberto, Rasgos Descriptores para el Reconocimiento de Objetos, Instituto Politécnico Nacional, México (2006).
9. Pajares Gonzalo, De la Cruz Jesús, Visión por computador, Imágenes digitales y aplicaciones, Alfaomega Grupo Editor (2004).
10. Adobe Photoshop CS2, versión 9.0, Adobe Systems Inc. (2005).
11. Umbaugh, S. E. Computer Vision and Image Processing: A Practical approach using CVIP tools. Prentice-Hall, EngleWood Cliffs: NJ (1998).

Non-Rigid Image Registration based on Form Transformations

Daniel Pimentel, María Elena Algorri, José Luis Mosso*, Gerardo Arrellín*

Instituto Tecnológico Autónomo de México, Sistemas Digitales
Río Hondo No. 1, Progreso Tizapán, 01080 D.F., México
danielpimentelalarcon@gmail.com, algorri@itam.mx
*Escuela de Medicina, Universidad Panamericana Mixcoac, México D.F.

Abstract. Image registration is an important tool for clinical diagnosis and treatment. Its objective is to find a geometrical transformation that allows the precise alignment of two different images of the same anatomical scene so that their information can be mutually enriched. The image registration method reported in this paper establishes a correspondence between regions in the images based on their form. The registration starts with a rigid alignment using affine transformations followed by a non-rigid transformation based on a thin-plate spline. We registered two MRI thoracic studies of a rabbit, and a set of MRI, PET and SPECT studies of a human brain. Experimental results were evaluated using Mutual Information as similarity criterion.

Keywords: Rigid and non-rigid image registration, thin-plate splines.

1 Introduction

Medical image registration is needed to compare or to integrate the information of different images of the same anatomical scene. The images can present variations due to changes in patient position, acquisition times, nature or calibration of the sensors (MR, PET, etc.), anatomical functioning (breathing, digestion, etc.), evolution of pathologies, or even because they were taken from different patients.

Image registration is the process of aligning two images, transforming one of them (floating image), to maximize its similarity with respect to the other (reference image), so that they can be related, compared and their information complemented.

Since registered images have a common reference frame, it becomes easier to monitor subtle changes in size or intensity in anatomical structures. Other applications of image registration are: To complement the information of multimodal images, to monitor anatomical changes through time or though medical treatment, to identify lesions, to make 3D reconstructions, for use in computer assisted surgery or even in statistical studies [5].

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 115-124

In fact, image registration has become an invaluable resource in biomedical research, particularly in neurosciences, where multimodal imaging has substantially contributed to the study of brain functioning [2].

Many image registration methods can be found in literature, with techniques varying, depending on the application and image type [1], from surface analysis [10,14], to evolutive computing [11,12] and neural networks [9]. The most popular methods are those based on measures of similarity [3,4,5,6], on geometrical features [7,8], and on domain transformations [13, 14, 15].

In this paper we present a non-rigid registration method based on geometrical features, particularly forms, that can register single and multimodal images with good precision. We show our results on two MRI axial studies of a rabbit's thorax (Fig. 1) as well as on a set of axial multimodal studies of a human brain. To evaluate our registration results we used the Mutual Information [16,17] of the images based on Shannon's entropy, before and after registration to measure the increase in image similarity. The results are presented in compact form using confidence intervals [20].

2 Method

The method reported in this paper uses a correspondence between the forms of regions in the images. The method transforms the forms of the regions in the floating image to adjust them to the forms of the regions in the reference image. The method consists of four stages: Preprocessing, feature extraction, preliminary rigid registration and non-rigid registration.

2.1 Preprocessing

Given the amount of noise present in the medical images that were used in the experiments, it was necessary to improve their quality before undergoing registration. Preprocessing of the images consisted of contrast enhancement, noise removal and edge detection. The algorithms used are described in [18]. The images shown in this paper have all been preprocessed.

2.2 Feature Extraction

After preprocessing, the method extracts the features that are common to the two images to be registered, so that a common coordinate frame can be established for both of them. The features used by the method are the exterior contour of the patient, the anatomical center and the main direction of the images (Fig. 1).

Exterior contour. To find the exterior contour of the images, these are first binarized to identify the coordinates of the foreground pixels with value $I(x,y)=1$. From these pixels we obtain the convex hull [32] using Graham's scanning algorithm [31]. The convex hull can present discontinuities, so it was necessary to complete the information of the contour represented by the convex hull using interpolation. We tried two different interpolation methods:

Splines: All the pixels of the convex hull were interpolated with a natural spline [33].

Nearest Neighbors: Although the convex hull is a closed form, sometimes consecutive pixels over it lie far away from each other in pixel space. This is the case of pixels over the convex hull that border concave sections of the contour, since such concave sections are far away from the convex hull. To better define the contour over such sections we drew an imaginary line over the distant (but consecutive) pixels of the convex hull and looked for the pixels of the binary image whose perpendicular distance to the imaginary line was shortest. Just a few additional pixels need to be found in this way, since then a new spline interpolation can be conducted over the pixels to complete the contour information.

Both methods are used distinctively depending on the concavity of each section of the contour: if the section is convex, interpolation with splines is used, if the section is concave we used a nearest neighbor search.

Anatomical center. Once the exterior contour is found, the anatomical center is calculated as the weighted average (ex, ey) of the coordinates of the contour.

Principal direction. By finding the principal direction of the reference and floating images, a common orientation can be calculated for them. Depending on the images, the methods to find the principal direction can vary considerably, from determining the longest axis of the image contours, to looking for specific patterns in the represented anatomy. Finding a common orientation for two very different images can be prone to errors, and in the worst case, when the direction cannot be determined automatically, the images can be annotated, or the principal direction determined interactively by a variety of methods [9], until the optimal direction is found when the Mutual Information of the registered images is maximized.

In our method, the longest axis of the contour was used as the principal direction of the reference images. However, due to the circularity of the floating images and to their high noise content, it was necessary to annotate them manually by clicking on the position of the rabbit's vertebrae in every image. The principal direction was then established as the vector passing through the anatomical center of the images and through the position of the vertebrae.

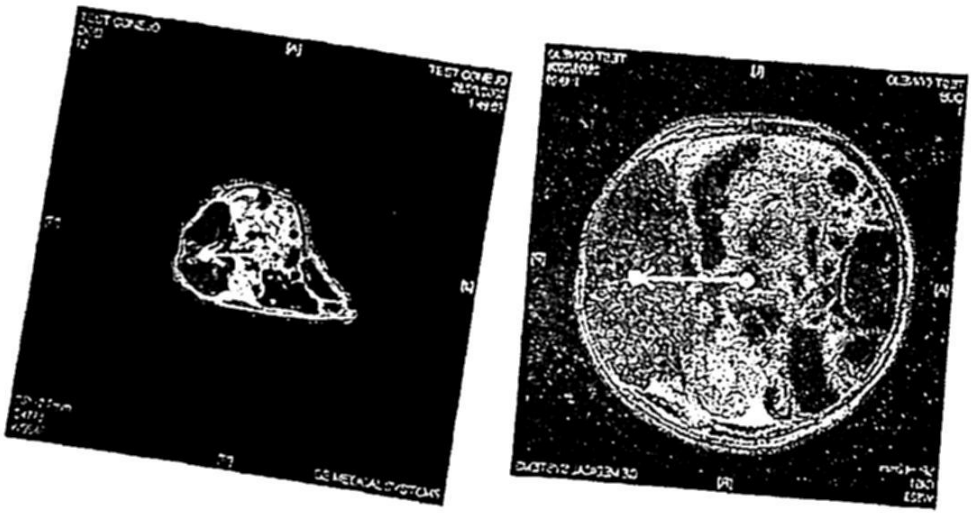
2.3 Preliminary Rigid Registration

Rigid registration is a necessary step to align the images based on their anatomical center and principal direction. Rigid registration gives both images a common anatomical center and a common principal direction as seen in Fig. 1, and it avoids unnecessary processing

during non-rigid registration [1]. Rigid registration is carried out using a translation and a rotation matrix [21].

2.4 Non-rigid Registration

For the non-rigid registration, a correspondence relationship has to be established between the points that form the contour of the reference image F_R, and the points that form the contour of the floating image F_F. There are various ways of establishing a correspondence between the points of the two contours. A simple one consists of associating every point along the reference contour with another point along the floating contour following a given criteria. We use this approach as follows: Let p_i be a point of F_R and e the common anatomical center of both images after rigid registration. Let l_i be the line that passes through e and p_i with equation: $y = m_i x + b_i$.



a) b)

Fig. 1. (a) Reference image after rigid alignment. (b) Floating image after rigid alignment.

To determine the exact point q_i on F_F that corresponds to p_i we look for the intersection of l_i and F_F (Fig. 3). We find the exact intersection point p_i using a line-line intersection strategy. The first line is l_i and the second line is the line formed by two consecutive points (x_i,y_i) (x_{i+1},y_{i+1}) over F_F that intersects l_i at a point (x,y) lying along the path between (x_i, y_i) and (x_{i+1}, y_{i+1}). Since we are dealing with discrete images, it is possible that q_i does not exactly belong to the finite set of points in F_F, in which case an interpolation is necessary.

Thin-plate-spline Deformation. Thin-plate splines [22] are bivariate interpolation functions based on physical principles that can be used to interpolate an arbitrary set of coordinates $(x,y,f(x,y))$. These splines are a generalization of the natural cubic spline for one dimension. They represent the minimal deformation that allows the coordinates of a

metal sheet (x_i,y_i) to adjust themselves to the new set of coordinate values (x_i', y_i') that is specified by the correspondence of contours. The idea is to model the floating image as a metal sheet, and make the contour points in it adjust themselves exactly to their corresponding points on the contour of the reference image. The rest of the pixel values $f(x_i,y_i)$ in the floating image are interpolated to adjust them to the new coordinate values specified by the correspondence. We must then find the function f(x,y) that minimizes:

$$\iint \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + \left(\frac{\partial^2 f}{\partial xy}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 dxdy \ , \tag{1}$$

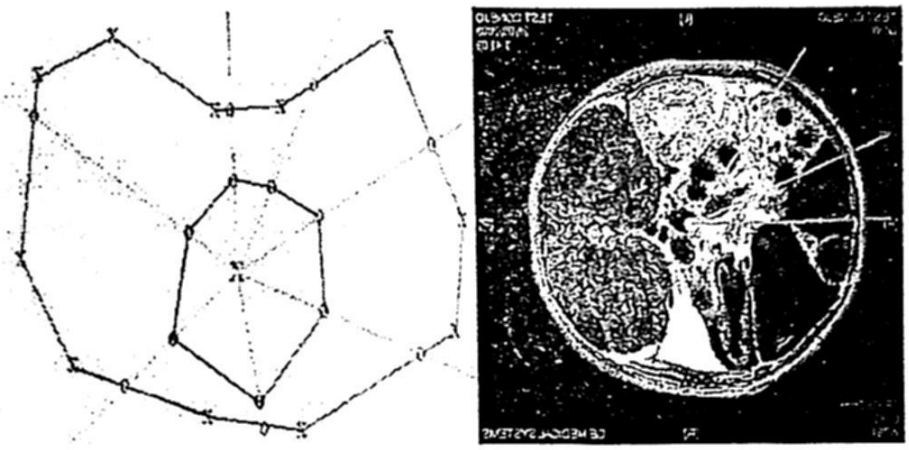


Fig. 2. (a) Method used to determine the correspondence of points in both contours. The anatomical center is represented by the middle X. The circles along the inner contour are the points that form the reference contour, the X's on the exterior contour are the points that form the floating contour. To find the points over the floating contour that correspond to the points on the reference contour, a ray-line is traced from the common center that passes through each point in the reference contour and intersects the floating contour. The exact intersection point is found using a line-line intersection algorithm. The circle points along both contours are then associated and will determine the deformation that the floating image will undergo. b) The correspondence method between both contours is illustrated in two MR images of the rabbit.

commonly known as the energy deformation function of the thin-plate spline. This function is minimized through the solution of a system of linear equations:

$$f(x,y) = a_0 + a_x x + a_y y + \sum_{i=1}^{n} \omega_i U(|x-x_i, y-y_i|) \ , \tag{2}$$

where:

$$U(t) = r^2 \log(r) \tag{3}$$

is the well known fundamental solution to the bi-harmonic equation $\Delta^2 U = 0$, that satisfies the condition of minimum energy deformation, $|\cdot|$ indicates the length of a vector, and the coefficients a_0, a_x, a_y, and ω_i are determined by the requirements of an exact interpolation, such that by the proper choice of ω_i, $f(x,y)$ maps the intensities of the pixels

in the floating image from their original coordinate space to the coordinate space specified by the correspondence of both images. Thin-plate splines provide a soft interpolation for derivatives of any order, they do not require manual specification of parameters, they offer a closed solution and a physical explanation of their energy function. By applying the thin-plate spline deformation to the floating image, we assign the intensity values of the pixels on its exterior contour, to the corresponding pixels on the exterior contour of the reference image, carrying out a bivariate interpolation of all the other pixel values on the floating image and applying them to their new positions on the reference image, to obtain the desired registered image as shown in Fig. 3.

3 Experimental Results

We acquired two MR studies, each of 20 images of the thoracic region of a rabbit. The first study was taken with the rabbit placed free in the scanner and using the full magnet that is used to take studies of adult patients. The second study was taken with the rabbit inside a smaller magnet usually used to take images of a hand or foot. The images had a resolution of 512 x 512 and a spatial separation of less than a centimeter. The images of the first study were established as reference images (Fig. 1a), and the images of the second study were the floating images (Fig.1b).

In an attempt to obtain a quantitative evaluation of the registration results, we used Mutual Information as a similarity measure of the registered images [23-30]. Statistically, Mutual Information can be interpreted as the dependency measure between two sub experiments X and Y, having two sampling subspaces S_x and S_y.

$$\left.\begin{aligned} S_X &= \{x_1, x_2, x_3, \ldots, x_m\} \\ S_Y &= \{y_1, y_2, y_3, \ldots, y_n\} \end{aligned}\right|, \tag{4}$$

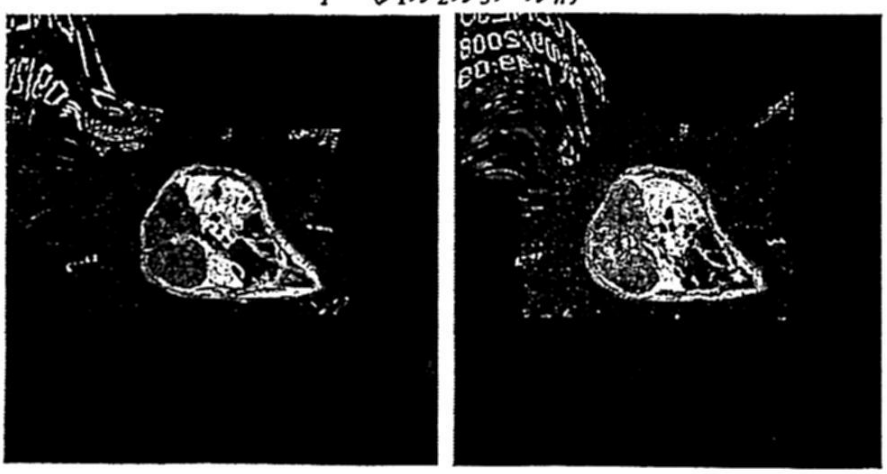


Fig. 3. Registered images.

With $p(x_i)$ and $p(y_i)$ the probabilities of events x_i and y_i respectively. To evaluate the similarity between the reference and the floating images, we consider the reference image,

sub experiment X, and the floating image, sub experiment Y, and we define the sampling subspaces as the possible intensity values of the pixels in the images. Mutual information is defined as:

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y \mid X) \\ &= H(X) + H(Y) - H(X,Y) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} p(x_i,y_j)\log\left(\frac{p(x_i,y_j)}{p(x_i)p(y_j)}\right) \end{aligned} \quad (5)$$

where $H(Y)$ is Shannon's entropy[17] of Y, $H(Y|X)$ is the entropy of Y given X, and $H(X,Y)$ is the joint entropy of X and Y.

When the images have minimum similarity, that is, when they are independent events, the mutual information between I_R and I_F (reference and floating images) is minimal, and when the images are completely dependent from each other, their mutual information is maximal.

We present here the results of measuring the mutual information between pairs of images in terms of 95% confidence intervals using the t-Student distribution [20] for three cases: OMI (original mutual information of the images before registration) as $I(I_R;I_F)$, RMI (mutual information after registration) as $I(I_R;I_R)$, and MMI (maximal mutual information $H(I_R)$. Note that MMI implies that both images are the same, meaning they have no complimentary information, which is a non-desired outcome and is shown here just for comparison. The results obtained are shown in Table 1 where it can be seen that, for 95 out a 100 registered images, their mutual information increased from 0.0406 to 0.2519, and from 0.0546 to 0.2893, that is a 5-fold increase.

Tabla 1. 95% Confidence Intervals

Measurement	Confidence Interval
$I(I_R,I_F)$	$P(0.0406 \leq OMI \leq 0.0546)=0.95$
$I(I_R,I_T)$	$P(0.2519 \leq RMI \leq 0.2893)=0.95$
$H(I_R)$	$P(0.9628 \leq MMI \leq 1.1370)=0.95$

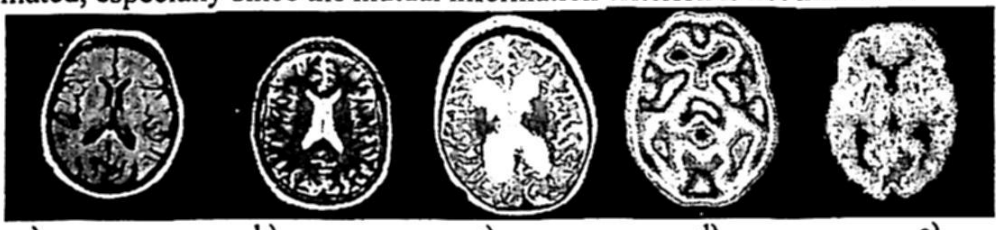
Although the conventional notation is to express confidence intervals in terms of probabilities, the correct interpretation is that if an experiment were realized once and again, in 95% of times, the value of mutual information would be between the limits of the confidence interval for each case.

4 Analysis

Most image registration methods propose iterative searches throughout the method to obtain the best registration parameters. In the method used in this paper, the parameters to use are only those of the preprocessing stage plus the principal direction of the images. The required preprocessing parameters are the size of the averaging filters, the degree of

membership in the fuzzy logic algorithms, the noise removal and binarization threshold values. The direction of the images can vary from 0° to 360°, and, if impossible to determine automatically, it can be found through an iterated search where the direction angle is varied in the floating image, until the best registration is achieved.

The variability of the parameters to achieve the best possible registration depends largely on the amount of noise present in the images. Proof of this are the reference images which are almost noise free and where the parameters were never changed, and even if they were slightly modified, the results showed almost no variation. On the other hand, for some of the very noisy floating images only a small set of parameter values resulted in adequate registrations. These successful parameter values were determined iteratively looking for registration results that provided the largest increase in mutual information. Noise also makes the process of determining the direction of the images, prone to error. And even the process of iteratively looking for parameter values can be contaminated, especially since the mutual information criterion is not infallible.



a) b) c) d) e)

Fig. 4. (a) MR reference image, (b-e) MR, MR, PET, SPECT floating images

We wanted to further test the flexibility of the reported method and its applicability to other types of medical images, so we did an experiment on brain axial images. We downloaded 5 multimodal brain studies (Fig.4): three MRIs, one SPECT and a PET. The results of this image registration experiment are shown in Fig. 5.

Fig. 5. Registered images.

5 Conclusions

Experimental results show that the non-rigid registration method reported in this paper can achieve high precision results while maintaining flexibility of application. As in all image-processing tasks, noise is a big issue. The success of the image registration method largely depends on the success with which noise is removed from the images. Noise is the

main cause for parameters being manually or iteratively adjusted. Besides noise, determining the principal direction of the images was the biggest challenge, and if some a-priori information can be used to precisely determine it without manual intervention, then the method becomes a good candidate for real time unsupervised registration.

Mutual information is a measure of similarity that, while not infallible, offers a solid similarity measure that allows the user to choose the best registration results especially when parameters must be determined iteratively.

Many areas in image processing can complement the results of image registration. Some examples are automatic image segmentation and classification, whose results can then lead to very precise registrations [1]. Our future work is focused on adding more specialized image processing tasks to the registration pipeline to increase its precision, flexibility and applicability.

References

1. Gottesfeld B. L., "A survey of image registration techniques", ACM Computing Surveys Vol. 24, Num. 4, 1992, pp. 325-376.
2. Joseph V. Hajnal, Derek L.G. Hill, David J. Hawkes, Medical Image Registration, CRC Press LLC, 2001.
3. R. Oeuvray and M. Bierlaire, "A new derivative-free algorithm for the medical image registration problem", International Journal of Modelling and Simulation, Vol. 27, Num. 2, 2007, pp. 115-124.
4. Shaoyan Sun, Liwei Zhang, and Chonghui Guo, "Medical Image Registration by Minimizing Divergence Measure Based on Tsallis Entropy", International Journal of Biomedical Sciences, Vol. 2, Num. 2, 2007, pp. 75-80.
5. Xiu Ying Wang, David Dagan Feng, Jesse Jin, "Elastic Medical Image Registration Based on Image Intensity", Int. Journal of Imag. and Graphics, Vol. 5, Num. 2, 2005, pp. 351-369.
6. S. Korukonda, MC. Prakash, R. Venkatachalam, and S. Ramasway, "Information theory based image registration", Review of Quantitative Nondes-tructive Evaluation, Vol. 26, 2007, pp. 531-538.
7. Hai-Ping Ren, Hu Yang, Bing-Rong Ma, Sheng-Zu Chen, andWen-KaiWu, "A novel magnetic resonance-positive emission image registration based on morphology", Bio-Medical Materials and Engineering, Vol. 13, 2003, pp. 187-196.
8. Jianhua Xuan, Yue Wang, Matthew T. Freedman, Tulay Adali, and Peter Shields, "Nonrigid Medical Image Registration by Finite-Element Deformable Sheet-Curve Models", International Journal of Biomedical Imaging, Vol. 2006, 2006, pp. 1-9.
9. Lifeng Shang, Jian Cheng Lv, Zhang Yi, "Rigid medical image registration using PCA neural network", Neurocomputing Vol. 69, 2006, pp. 1717-1722.
10. Berthold K. P. Horn et. al, "Using Synthetic Images to Register Real Images with Surface Models", Communications of the ACM, Vol. 21, Num. 11, 1978, pp. 914-924.
11. Flávio Luis Seixas, Luiz Satoru Ochi, Aura Conci, Débora C. M. Saade, "Image Registration Using Genetic Algorithms", GECCOí08 ACM, Julio 2008, pp. 1145-1146.
12. Ju Han and Bir Bhanu, "Hierarchical Multi-sensor Image Registration Using Evolutionary Computation", GECCO í05 ACM, 2005, pp. 2045-2052.

13. Tarek A. El-Ghazawi, Prachya Chalermwat, Jacqueline Le Moigne, "Wavelet-Based Image Registration on Parallel Computers", ACM, 1997.
14. Q. P. Zhang, M. Liang, and W.C. Sun, "Multi-resolution image data fusion using 2-D discrete wavelet transform and self-organizing neural networks", ACM, 2004, pp. 297-301.
15. M. H. Malik, S. A. M. Gilani, and Anwaar-ul-Haq, iAdaptive Image Fusion Scheme Based on Contourlet Transform and Machine Learningî, International Review on Computers and Sortware (I.RE.CO.S), Vol. 3, Num. 1, 2008, pp. 62-69.
16. Jan C A van der Lubbe, Information Theory, Cambridge University Press, 1997.
17. Roberto Togneri, Christopher J.S. deSilva, Fundamentals of Information theory and coding design, CRC Press LLC, Nueva York, 2000.
18. K. J. Ray Liu, Pattern Recognition and Image Preprocessing, 2a Edición, Board, 2002.
19. Rafael C. Gonzalez, Richard E. Woods, Dig. Image Processing, Pearson PH, 3rd. Ed., 2008.
20. Devore, J, Probability and Statistics for Engineering and the Sciences, 3rd Ed, Wadsworth Publishing Company, Califoria, 1982.
21. Stanley I. Grossman, Álgebra lineal, McGraw-Hill, México, 1999.
22. Bookstein, Fred L., "Principal Warps: Thin-Plate Splines and the Descomposition of Deformations", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 6, 1989, pp. 567-585.
23. Amira Serifovic-Trbalic, Damir Demirovic, Naser Prljaca, and Zekerijah Sabanovic, "Evaluation of Interpolation Methods in Mutual Information-Based Medical Image Registration", Original Paper, Vol. 15, Num. 4, 2007, pp. 191-196.
24. Jiangang Liu and Jie Tian, "Registration of Brain MRI/PET Images Based on Adaptive Combination of Intensity and Gradient Field Mutual Information", International Journal of Biomedical Imaging, Volume 2007, 2007, pp. 1-10.
25. C. Fookes, M. Bennamoun, "Rigid Medical Image Registration and its Association with Mutual Information", International Journal of Pattern Recognition and Arti.cial Intelligence, Vol. 17, Num. 7, 2003, pp. 1167-1206.
26. Edward J. R. Somer, Paul K. Marsden, Nigel A. Benatar, Joanne Goodey, Michael J. O.Doherty, Michael A. Smith, "PET-MR image fusion in soft tissue sarcoma: accuracy, reliability and practicality of interactive point-based and automated Mutual Information techniques", Europ. Journal of Nuclear Med. and Molecular Imag., Vol. 30, Num. 1, 2003.
27. Philippe Thévenaz, Michel Bierlaire, Michael Unser, .Halton Sampling for Image Registration Based on Mutual Information., Sampling Theory in Signal and Image Processing, Vol. 7, No. 2, 2008, pp. 141-171.
28. Charlse R. Meyer, Bradford A. Mo□at, Kyle K. Kuszpit, Peyton L. Bland, Paul E. Mckeever, Timothy D. Johnson, Thomas L. Chenevert, Alnawaz Rehemtulla, and Brian D. Ross, "A Methodology for Registration of a Histological Slide and In Vivo MRI Volume Based on Optimizing Mutual Information", Mol. Imaging, Vol. 5, Num. 1, 2006, pp. 16-23.
29. L. Thurfjell, Y. H. Lau, J. L. R. Andersson, B. F. Hutton, "Improved efficiency for MRI-SPET registration based on Mutual Information", European Journal of Nuclear Medicine, Vol. 27, Num. 7, 2000, pp. 846-856.
30. N. Cvejic, C. N. Canagarajah, and D. R. Bull, Image fusion metric based on Mutual Information and Tsallis entropy., Electronics Letters, Vol. 42, Num. 11, 2006.
31. Graham, R. L., "An efficient algorithm for determining the convex hull of a planar set" Inform. Prec. Letlers 1, 1972, 132-133.
32. Bradley Efron, "The convex hull of a rand. set of points", Biometrika, 1965, pp. 331-343.
33. Hsieh Hou, Andrews, H., "Cubic splines for image interpolation and digital filtering", Acoustics, Speech and Signal Processing, IEEE Transactions, Vol. 26, 1978, pp. 508-517.

Pattern Recognition Techniques

Efficient Pattern Recognition of Dynamic Structures Formed by Multiagent Systems in Soccer Robotics Domain

Huberto Ayanegui§ and Fernando Ramos‡

§ Universidad Autonoma de Tlaxcala
Facultad de Ciencias Basicas, Ingenieria y Tecnologia
Calzada de Apizaquito s/n, 90350,
Apizaco, Tlaxcala, Mexico.
‡ Tecnologico de Monterrey Campus Cuernavaca
Autopista del Sol, Km. 104, Col. Real del Puente.
Xochitepec, Morelos, Mexico. 62790
{huberto.ayanegui, fernando.ramos}@itesm.mx

Abstract. This work is focused on the recognition of team patterns represented by different formations played by a soccer team during a match. In the soccer domain, the recognition of formation patterns is difficult due to the dynamic and real time conditions of the environment as well as the multiple interactions among team mates. In this work, some of these multiple interactions are modeled as relations represented by a topological graph which is able to manage the dynamic changes of structures. Thus, the topological graph serves to recognize apparent changes of formations from real changes of them. The proposed model has been tested with different teams in different matches of the Robocup Simulation League. The results have shown that the model can recognize the different main formations used by a team during a match even the multiple changes of the players due to the dynamic nature of a match.

Key words: Pattern recognition, robotic soccer, formations, dynamic behavior.

1 Introduction

Formations are the way a soccer team lines up its defense, midfield, and attack line during a match. When talking about formations, defenders are listed first and then midfielders and forwards. For example, a code 4:4:2 represents a formation composed by four defenders, four midfielders, and two forwards. As in the real soccer game the goalkeeper is not considered as part of the formation. Certainly, the dynamic conditions of the soccer game difficult the task of building adequate representations able to facilitate the recognition of formation patterns.

Usually, teams playing in strategic and organized ways search for respecting predefined patterns or formations [5][8]. The purpose of this work is the recognition of patterns represented by formations played by a team during a match. If

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 127-135

the recognition task is performed on an opponent team, a soccer team can obtain advantages over it [2]. Visser and colleagues [8] recognize formations of opponent teams using neural networks. This work feeds the observed player positions into a neural network and tries to classify them into a predefined set of formations. If a classification can be done, the appropriate counter formation is looked up and communicated to the players. Due to the fact that Visser does not represent the multiple relations between players, so the neural network has a low level of accuracy for some cases of soccer teams.

Riley and colleagues in [7] use a model to identify 'home areas' of players to recognize formations. A home area specifies the region of the field in which the agent should generally be. Thus, they propose that identifying home areas, the agents can infer a role in the team (defender, midfielder or forward players). A drawback of this approach is that due to dynamic conditions of the world, the player movements can generate such a wide range extending considerably the home areas, which difficult the task of determining the role of a player, therefore a correct formation.

Kuhlmann and colleagues [5] learn team formations similar to Riley and Veloso [7], using home positions. They model the formation as a home position (X,Y) and introduce a ball attraction vector (BX,BY) for each player. The X and Y values are calculated as the average x and y coordinates of the observed player during the course of the game. Values for BX and BY were handpicked for each position and were found through brief experimentation. A weakness of this work is that the home positions have to be adjusted, for some cases, manually.

It is presented in this work an efficient model to recognize patterns of formations based on a representation that takes into account multiple relations among defender, midfielder and forward players. The test domain for this research is simulated robotic soccer, specifically, the Soccer Server System [6], used in the Robot World Cup Initiative [4], an international AI and robotics research initiative. The system is a rich multiagent environment including fully distributed team agents in two different teams composed of eleven agents.

2 The Multiple Relation Model

The focus of this work is on teams that play following patterns of high level of abstraction based on a distribution of zones named Defensive (D), Middle (M) and Attack (A), as in classic soccer game. These patterns will be represented as follows: D:M:A. Due to the dynamic conditions of the soccer game, the players are in constant movement and temporally breaking the alignment of players belonging to a zone. To handle the constant changes without an expressive representation of the relations between players can result in an inefficient way of recognizing formations submitted to a dynamic environment. In the next sections it will be explained how the zones and the players belonging to them are recognized.

2.1 Recognition of Team Zones

As in human soccer domain the players in robotic soccer should tend to be organized [3]. That is, each player has a strategic position that defines its movement range in the soccer field. The role of a player is quite related with a predefined area within which an individual player can play basically in the field. Any behaviours of a player depend on its current role. According to the position of the player, roles in robotic soccer can be divided into four types: goalkeeper, defenders, midfielders and forwards. Different roles are associated with different positions and different behaviours that players assume. However, due to the dynamic changing conditions of a match, a defender could become a forward temporarily as his team is trying to attack. As well, a forward could become a defender temporarily when his team is being attacked. So the roles of a player are dynamically changing and a player can have dynamic behaviours in a match. Consequently, the recognition of formation patterns is difficult due to the dynamic and real time conditions of the environment. It is needed to determine, first of all, the belonging of players to a specific zone. This first approach tells us the clustering of players to a zone. In this work, the clustering algorithm, K-means, is applied to meet this first stage. K-means works well on large data sets due to its linear time complexity. K-means classifies a given data set through a certain number of clusters (assume k clusters) fixed a priori. In this work, k=3 such that three zones will be defined: defensive, middle and attack zones (D:M:A). From the log file (game film), the data from one team is extracted and K-means is applied in each simulation cycle of the game. The positions of each player, with respect to the x axis, are taken as the input of the clustering algorithm and the output of clustering is the classification, according to their x position, of all players of the team in the three clusters. Due to the continuously moving of players, it is not feasible to conclude what players are in each zone of a team from a single cycle only, a period of time should be considered.

Clustering algorithm is useful to determine the three zones of a team but it is not able to represent the multiple relations between players of each zone. Given that patterns of formations are based on relations that determine structures then an additional model is crucial for the recognition of patterns of formations. Such model has to be sufficiently robust in order to manage the constant positional changes of players. The next section describes how a set of triangular sub-graphs connected together, that uses as input the result of K-means algorithm, can be used as an adequate representational model able to facilitate the recognition of formation patterns.

2.2 Representation of multiple relations

A formation is represented by a set of relations between players. Thus, the relations represent the structure that supports a formation. So, a change of relations between players entails a change of formation. It is needed at least the change of one relation to transform one structure into another one. Constant changes of relations could occur because the multiple relations in a formation and the

dynamic nature of a match. Figure 1(a) illustrates the relations of each one of the players with the rest of their teammates. A total of 90 relations are given by the formula: $n(n-1)$, where n represents the number of players. This formula considers two relations by each pair of players. Thus, one relation is represented by the link from player A to player B and the second one from player B to player A. For practical reasons it is considered just one of these relations. Thus, the total of relations is $\frac{n(n-1)}{2} = 45$. Figure 1(b) illustrates these 45 relations.

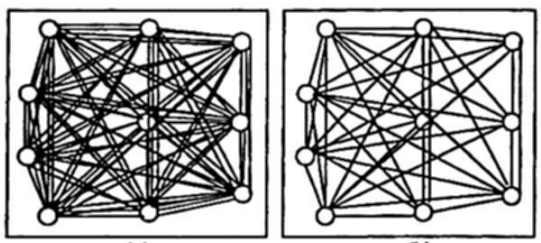


Fig. 1. All posible relations between players of a soccer team. (*a*) 90 relations and (*b*) 45 relations

On the one hand, the control of such number of relations becomes very difficult to be managed because any change of relations would produce a change of structure. In addition, it could happen that several changes of relations occur at the same time. Then the problem of detecting what relations are provoking changes of structures becomes much more difficult to be managed. On the other hand, the 45 relations are not relevant in a real match, because a relevant relation is the one in which a player uses to exchange passes and positions in a strategic way. Thus, a player stays related with his closer neighbor belonging to his zone and the closer neighbor belonging to the neighbor zone. In this work, the goal is to build a simple but robust structure based on relevant relations able to manage the dynamic nature of the game in a topological way, based on triangular sub-graphs that are built as indicated below:

- *Step 1.* Let $Nz_i = \{a_i < b_i < c_i < ... < n_i\}$ be finite number of nodes (players) belonging to a zone Z_i, where player a_i has coordinates located in the top of the zone and player n_i at the bottom of it. The rest of the nodes are located at intermediate positions. Thus, a_i is linked with b_i, b_i is linked with c_i, and so on (see Fig. 2(a)).
- *Step 2.* Once the nodes of the zones have been linked, nodes of neighbor zones based on minimal distances are linked until a planar graph is built (see Fig. 2(b)).

Figure 2(c) shows the planar graph represented by triangular sub-graphs as result of applying the previous two steps.

The total number of relations of a graph, which has been built based on the method described above, is given by $N_m + 15$; where N_m is the number of

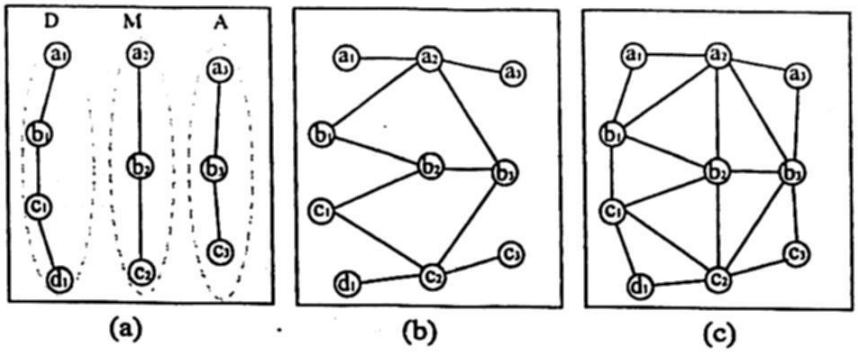


Fig. 2. (*a*) *Step 1.* Neighbor nodes of the same zone are linked. (*b*) *Step 2.* Neighbor nodes of neighbor zones are linked. (*c*) Planar graph obtained from step 1 and step 2.

nodes of the middle zone (Due to the lack of space the deduction of this formula is not described in this work). For instance, for a formation 4:4:2, the number of relations will be 19, because $N_m = 4$. The advantages of this method are expressed below:

- The number of relations has been reduced from 45 to 19 for the formation 4:4:2. Then, 26 relations have been eliminated.
- In a formation the minimal number of relations of a node is 2 meanwhile the maximal number is 9. An example of this kind of formation is 9:1:0. For this formation the total number of relations is 16.
- Triangular subgraphs are able to assume a topological behavior. That is, even if a structure is deformed because positional changes of nodes, the topological property of the triangular graphs helps to preserve the structure.

3 Pattern Recognition Process

Fig. 3 shows the process to recognize patterns of formations and changes of structures that support the formations. The first module serves to determine the zones by using a clustering algorithm; the second module builds the multiple relations which are expressed by a topological graph and finally in the third module the changes of structures are detected if topological properties of a defined structure have been broken.

Module 1. *Recognition of team zones.* The algorithm of clustering is performed during the first cycles of the match and it is stopped when the number of players in each group does not change. In this way, the three zones of a team, defensive, middle and attack zones are recognized.

Module 2. *Building multiple relations and a topological graph.* Based on the three zones recognized by the clustering algorithm, relevant multiple relations and a topological planar graph are built.

Module 3. *Recognition of Changes of Structures that support Team Formations.* Changes of structures are detected if topological properties of a defined

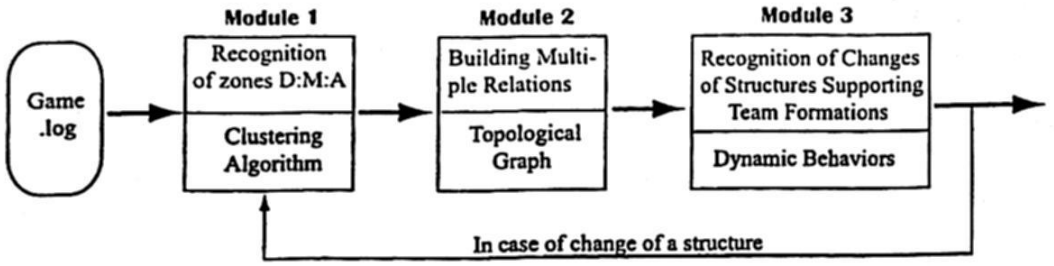


Fig. 3. Process to recognize pattern formations.

structure have been broken. A graph G is planar if it can be represented on a plane in such a way that the nodes represent different points and two edges should be encountered only at their ends. The intersection of two edges out of their ends breaks the planar property of the graph G. This graph G is also named as planar topological graph [1]. In a planar graph any pair of nodes can be linked. In addition, any link of the graph should not be intersected by any other link. Otherwise, if the topological property of the graph has been broken then another structure supporting a formation should be built. An example of intersected links is shown in Figure 5(b). Intersections occur when players change their roles in order to build a new formation or due to reactive behavior in response to the opponent. If there are intersections of links occurring in a short period of time (for example, 50 or less cycles), then they have not considered as a change of formation. But, if they occur during a long period of time (for example, 50 or more cycles), clustering algorithm should redefine the zones and a new topological graph should be built.

4 Experimental Results

In this section, important experimental results are shown. The results to be shown are derived from a match of the FC Portugal soccer team of the RoboCup Simulation Championship. This team has won several world RoboCup championships. However, in order to validate the approach presented in this work a vast number of matches has been analyzed.

Fig. 4 shows the results of the clustering algorithm derived from the first 1500 cycles of a match. As you can see, the clustering algorithm recognized the three zones of a team. 10 different classifications (formations) have been detected. However, formation 4.3:3 has been recognized more times than others. In particular, it has been recognized most frequently during the first 1000 cycles. The formation 4:2:4 was recognized in the next 500 cycles. On the contrary, formations such as 3:5:2, 2:4:4, 5:2:3, and 5:3:2 have been detected during a very short period of time.

Fig. 5(a) shows the 9 triangles of the planar graph of some simulation cycles during the first 1000 simulation cycles. It is observed from this sequence of graphs that the structured supporting the formation 4:3:3 is maintained. How-

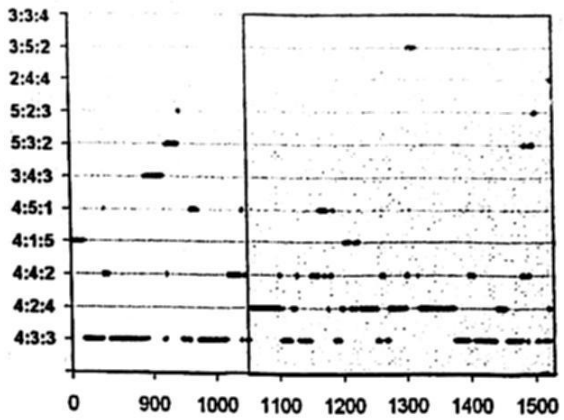


Fig. 4. Clustering results of the first 1500 simulation cycles

ever clustering algorithm recognized different formations even that the relations between agents have not changed. The team started with the formation 4:3:3 and changed to 4:2:4 at the cycle 1050, just after scoring the second goal. A possible cause could be that the score was in favour of this team, so the team has possibly decided to experiment a more offensive behaviour. Fig. 5(b) shows intersections of relations that indicates a change of formation. Once intersections of links have been detected the relations are redefined and a new formation is determined as described by module 2 discussed before. Fig. 5(c) shows the new planar graph built because the intersections shown in Fig. 5(b).

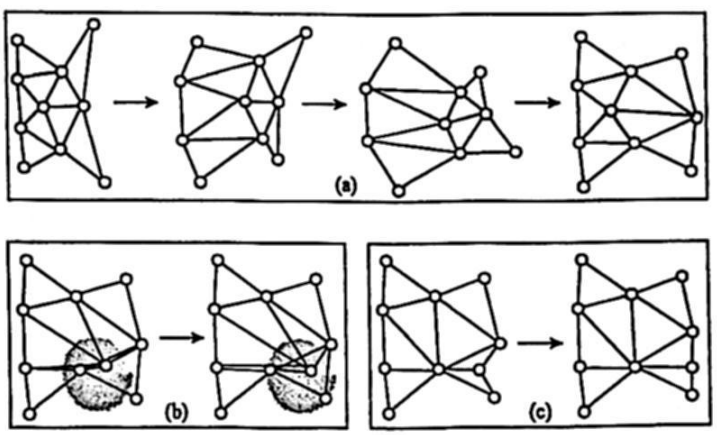


Fig. 5. (*a*) Four frames showing a planar graph during the first 1000 simulation cycles. (*b*) Intersection of relations (shadow region). (*c*) Building of a new planar graph for a new formation.

5 Conclusion

In this work, we have been concerned with the recognition of pattern dynamic behaviors within complex, competitive and real time domains, such as soccer robotic games. The patterns to be recognized are formations that can change because strategic reasons or due to reactive behavior in response to the opponent. These facts bring about multiple interactions between agents of a team, and make difficult the task of recognizing formation patterns. In this work, we presented an efficient model to recognize formations based on a rich representation that takes into account multiple relations among players, including the neighbourhood relation between players of the same zone and the neighbourhood relation between players belonging to neighbour zones. In particular, a topological model, based on triangular sub-graphs, has been built, which is able to manage the deformations of structures due to the dynamic changes and the multiple relations. A vast number of matches have been analyzed to test the model. The experiments have shown that the model is able to recognize formations and apparent and real changes of them.

As future work variants around a formation will be modelled. A variant is defined as a temporal change around a given formation during a short period of time. We consider that variants are, in most of the times, the strategic key of the teams that decide the final result of a match.

References

1. Claude Berge. *Graphes.* Guathier-Villars, 1983.
2. David Carmel and Shaul Markovitch. Incorporating opponent models into adversary search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, OR, 1996.
3. Gal A. Kaminka, Mehmet Fidanboylu, Allen Chang, and Manuela M. Veloso. Learning the sequential coordinated behavior of teams from observations. In Gal A. Kaminka, Pedro U. Lima, and Ral Rojas, editors, *RoboCup*, volume 2752 of *Lecture Notes in Computer Science*, pages 111–125. Springer, 2002.
4. Hiroaki Kitano, Milind Tambe, Peter Stone, Manuela M. Veloso, Silvia Coradeschi, Eiichi Osawa, Hitoshi Matsubara, Itsuki Noda, and Minoru Asada. The robocup synthetic agent challenge 97. In Hiroaki Kitano, editor, *RoboCup*, volume 1395 of *Lecture Notes in Computer Science*, pages 62–73. Springer, 1997.
5. Gregory Kuhlmann, Peter Stone, and Justin Lallinger. The ut austin villa 2003 champion simulator coach: A machine learning approach. In Daniele Nardi, Martin Riedmiller, Claude Sammut, and Jos Santos-Victor, editors, *RobuCup*, volume 3276 of *Lecture Notes in Computer Science*, pages 636–644. Springer, 2004.
6. Itsuki Noda and Ian Frank. Investigating the complex with virtual soccer. In Jean-Claude Heudin, editor, *Virtual Worlds*, volume 1434 of *Lecture Notes in Computer Science*, pages 241–253. Springer, 1998.
7. Patrick Riley, Manuela Veloso, and Gal Kaminka. An empirical study of coaching. In *Distributed Autonomous Robotic Systems 5*, pages 215–224. Springer-Verlag, 2002.
8. Ubbo Visser, Christian Drcker, Sebastian Hbner, Esko Schmidt, and Hans-Georg Weland. Recognizing formations in opponent teams. In Peter Stone, Tucker R.

Balch, and Gerhard K. Kraetzschmar, editors, *RoboCup*, volume 2019 of *Lecture Notes in Computer Science*, pages 391–396. Springer, 2000.

Optimizing the Compact Genetic Algorithm Design with Cellular Automata

Alejandro León-Javier[1], Marco A. Moreno-Armendáriz[1] and Nareli Cruz-Cortés[1],

[1] Centro de Investigación en Computación - IPN
Av. Juan de Dios Batíz s/n Unidad Profesional Adolfo López Mateos
Col. Nueva Industrial Vallejo, Mexico,D.F., 07738, Mexico.
ljavierb07@sagitario.cic.ipn.mx, {marco_moreno, nareli}@cic.ipn.mx

Abstract. The Compact Genetic Algorithms (cGAs) are searching methods used in different engineering applications by their simplicity and low consumption of resources when compared with other Evolutionary Algorithms. These characteristics make them very attractive to be implemented in hardware. In this paper we propose some manners of optimizing a cGA design in VHDL by using Cellular Automata (CAs) with the aim of implementing them on a FPGAs.

Keywords: Compact Genetic Algorithm, Cellular Automata, FPGA, VHDL.

1 Introduction

The Compact Genetic Algorithms (cGAs) are classified as Estimation of Distribution Algorithms (EDAs) or Probabilistic Model Building Genetic Algorithms (PMBGAs). They are searching methods that mimic the behavior of conventional Genetic Algorithms (GAs) with uniform crossover [1]. The cGAs replace the population with a probability vector, where each component of the vector is updated by shifting its value by the contribution of a single individual to the total frequency assuming a particular population size [2].

Due to their particular characteristics, the cGAs can be implemented in a hardware platform easier than other Evolutionary Algorithms. The cGAs offer notable advantages in resources consumption due to their intrinsic simplicity.

There exist some publications showing Evolutionary Algorithms implementations based on hardware platforms. For example, in [3] the authors proposed the design of a hardware-based architecture to perform a Genetic Algorithm on a FPGA, called FPGA-based Genetic Algorithm Kernel. In [4] was proposed a Genetic Algorithm based on a chip LSI that includes reconfigurable logic hardware, a memory maintaining the individuals and a training data memory in a 16-bits CPU core. In [5] a cGA design and its

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 137-144

implementation was presented by using VERILOG language on a Xilinx FPGA. Gallagher et al, proposed in [6] to add elitism, mutation, and resampling to a cGA. These features improved the quality of found solutions by the cGA. In addition to that, they proposed to use a modular design to implement this algorithm on micro-controllers using VHDL. In [7] the authors proposed a parallel cellular cGA topology which is implemented on a FPGA.

Despite the existence of diverse architectures for different Evolutionary Algorithms, the majority of them have some disadvantages, maybe the most remarkable is that they consume many hardware resources or computing time (although the computing time is less when compared against software versions, it is not enough for real-time applications). Then in this work, we propose a cost-reducing architecture of hardware resources while preserving a good velocity of convergence by using Random Number Generators based on Cellular Automata.

```
1) Initialize probability vector
     for i:=1 to l do PV[i]:=0.5;
2) Generate two individuals from the vector
     a:=generate(PV);
     b:=generate(PV);
3) Let them compete
     winner,loser:=compete(a,b);
4) Update the probability vector towards the better one
     for i:=1 to l do
        if winner[i]≠loser[i] then
            if winner[i]=1 then
               PV[i]:= PV[i]+1/n;
            else
               PV[i]:= PV[i]-1/n;
5) Check if the vector has converged
     for i:=1 to l do
        if PV[i]>0 and PV[i]<1 then
            return to step 2;
6) PV represents the final solution
Compact GA parameters:
n: population size.
l: chromosome length.
```

Fig. 1. Pseudocode of cGA [8].

2 The Compact Genetic Algorithm

The cGA processes a Probability Vector (*PV*) with length *l*, which is initialized to *(0.5, 0.5,... , 0.5)*. Next, the individuals *a* and *b* are generated according to the *PV*. Then,

the fitness value of these individuals is compared and, the individual with better fitness is named the *winner* and the other one is called the *loser*.
Now, if *winner[i]* ≠ *loser[i]*, then *PV[i]* will be updated as follows: if *winner[i]=1* then *PV[i]* will be increased by *1/n*, otherwise, *VP[i]* will be decreased by *1/n*. Note that if *winner[i]=loser[i]*, the *VP[i]* will not be updated. These statements are repeated until each *PV[i]* becomes zero or one. Finally, *PV* represents the final solution [5]. The pseudocode of the cGA is shown in Figure 1.

3 Design of a Compact Genetic Algorithm Based on Hardware

The proposed design will be implemented on a FPGA, therefore, we will use VHDL to create the cGA's modules. They will be able to be connected to process the *PV* and at the same time, to parallelize the existing modules. These modules are called components and they have been designed as follows.

Compact Genetic Algorithm (cGA). This is the main component. It is based on the other components. Here, we have the registers *PV0,PV1,PV2,..., PVk* which are bit arrays representing a unsigned integer data in a range [*0, 2^k-1*], being *k* the length of bit array. The *PV* will be an integer array because VHDL is not synthesize floating point data (unless one designs them) and we would need a Random Number Generator in floating point for individual's generation, making more complicated the design. Also it includes registers (*ind1* and *ind2*) to store the individuals generated according to *PV*. The registers *RN1_0, RN1_1,..., RN2_0, RN2_1,...* store the random numbers generated by *RNG* components (all *RNG* components work in parallel). The *cGA* component is conformed by the following components: Random Number Generator (*RNG*), Individuals Generator (*IndGen*), Fitness Evaluator (*FEv*), Probability Vector Updater (*PVU*) and Probability Vector Checker (*PVC*). All this components are synchronize by a Finite State Machine descrited later.

Random Number Generator (RNG). Having a function or component as Random Number Generator is essential for any heuristic algorithm. Although exist many ways of generating pseudo-random numbers, only few of them are feasible to be designed into hardware due to consumption of resources that they demand. One interesting manner of generating random numbers is by using Cellular Automata. The Cellular Automata are composed by memory cells and rules. The Cellular Automaton-based Pseudo-Random Number Generators (CAPRNG's) can be of any length (no necessarily 32 bits), that is, we could have sequences of 2^n-1 elements (being *n* the number of bits of CAPRNG) for which we only need a memory cell (a D-flip-flop synchronous with *set* and *clear*) and its corresponding state rule (a combinatorial function) for each bit. These generators emulate populations of 2^n-2 individuals (being *n* the number of bits of CAPRNG), this way we allow achieving a reduction of bits in the generator and probability vector.

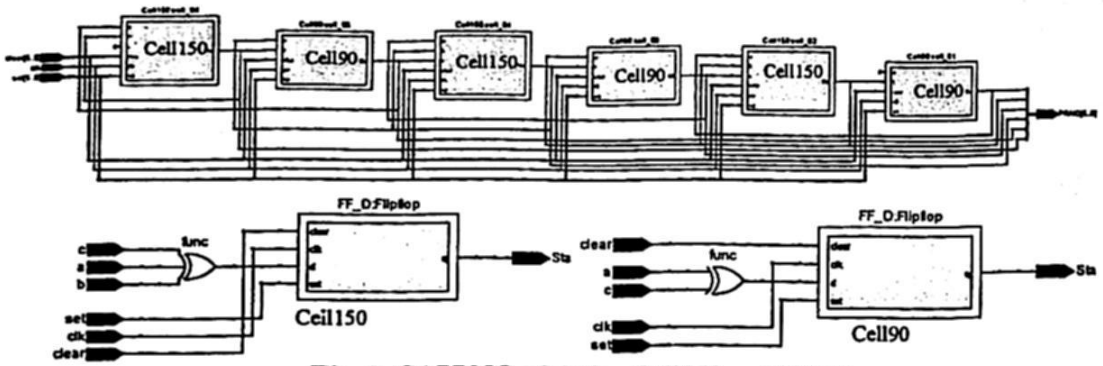


Fig. 2. CAPRNG of 6-bits, Cell150 and Cell90.

In Figure 2 we can notice that CAPRNG of 6 bits is composed of cells containing the rule 90 or 150 (Cell90 and Cell150 respectively). The rule 90 is: $a \oplus c$, and the rule 150 is: $a \oplus b \oplus c$. According to a selected cell, the pin *a* is connected with the pin *sta* of the left cell, the pin *b* is connected with the pin *sta* of the same cell, and pin *c* is connected with the pin *sta* of the right cell. In [9] it is shown how to design a CAPRNG's of 4 to 28 bits of length. For seeding the CAPRNG we should use the pins *set* and *clear* (for example, *set* array ='100101', *clear* array='011010') at the beginning and then, the pins *set* and *clear* will be cleared.

Individual Generator (IndGen). When this component is activated, it receives two random numbers and the *PV[i]* at a time. It generates one bit of each individual (*ind1[i]* and *ind2[i]*) in the i-th position at a clock cycle. The condition for generating one bit of a individual is: *if RN1≤PV[i] then ind1[i]='1', else ind1[i]='0'*, being *RN1* and *PV[i]* bit arrays of same length representing unsigned integers. All *IndGen* components contribute in generating two individuals in a clock cycle (working in parallel).

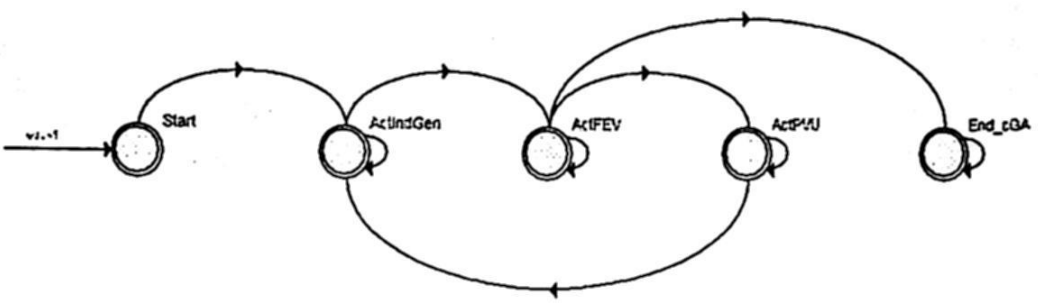
Fitness Evaluator (FEv). This component can be very variable because it contains the objective function, which can be very complex or very simple, depending on problem at hand. There exists the possibility that this component includes a Finite State Machines, where it accomplishes decoding and evaluation for floating point computing. The *FEv* component receives two bit arrays representing to individuals and it outputs a bit called *RES*. In general, first, we obtain both fitness values *fit1* and *fit2*, then we apply the next assignations: *if fit1≥fit2 then RES<='1', else RES<='0'*. where *RES* is a flag indicating which one of both individuals is the winner.

Probability Vector Updater (PVU). To update the *PV*, three data are received: the winner individual (*op*), the bit from the individual one on the i-th position (*ind1[i]*) and the bit from individual two on the i-th position (*ind2[i]*). The *oe* signal is the result of *ind[1] xor ind2[i]*. *PV[i]* is updated according to the next conditions:

```
if (oe='1' and ind1[i]='1') then
 if (op='1' and PV[i]<lim) then
   PV[i]<=PV[i]+1;
 elsif (op='0' and PV[1]>1)  then
   PV[i]<=PV[i]-1;
 end if;
elsif oe='1' and ind1[i]='0' then
 if (op='1' and PV[i]>1) then
   PV[i]<=PV[i]-1;
 elsif (op='0' and PV[i]<lim) then
   PV[i]<=PV[i]+1;
 end if;
end if;
```

lim is an integer value and it is equal to 2^n-*1* (being *n* the length of *PV[i]*). This limit is used to avoid overflow of *PV[i]*.

Probability Vector Checker (PVC). Due that each element of *PV* is an integer (or bit array), it is necessary to have other register (*SolPV*), where each element of *PV* be zero or one (final values). The *PVC* component updates each element of *SolPV* at each generation. This component activates a register called *RDY_PVC* when all elements of *PV* have converged. When *RDY_PVC=1*, the cGA has finished.



Transition conditions

	Current state	Next State	Rule
	Start	ActIndGen	
2	ActIndGen	ActIndGen	flag≠'0' and RDY_PVU≠"111"
	ActIndGen	ActFEV	flag='0' and RDY_PVU≠"111"
4	ActFEV	ActFEV	RDY_IndGen≠"111"
	ActFEV	ActPVU	RDY_IndGen="111" and RDY_PVC≠"111"
6	ActFEV	End_cGA	RDY_IndGen="111" and RDY_PVC="111"
	ActPVU	ActPVU	RDY_FEV≠'1'
8	ActPVU	ActIndGen	RDY_FEV='1'
	End_cGA	End_cGA	

Fig. 3. Finite State Machine (FSM) of the cGA component.

4 Synchronization by Finite State Machine

In order to synchronize the components previously proposed, we will use a Finite State Machine. A Finite State Machine is a sequential circuit with a finite number of states. The usage of this machine will guarantee the correct behavior of the whole design. The machine is defined by two functions, the first one calculates the next state of the system and the second one the output [10].

We use a Mealy Machine, which uses a clock as synchronizing signal for transitions. Our Finite State Machine is showed in Figure 3.

The FSM use some registers as *RDY_PVU*, indicating that *PVU* components have finished their execution if it is equal to *"1111..."*, otherwise, they have not finished. The same nomenclature is used for other components. Another register used is *ActIndGen*, which activate the *IndGen* component for generate the individuals. The same nomenclature is used for activate other components. All the transition conditions that synchronize the FSM are showed in Figure 3. The *RNG* is activated when *FEv* is activated. In *Start* state, the *RNG* component is seed.

5 Experiments and results

In our experiments, almost all registers of the cGA were sized with the same length (typically *PV* length). To evaluate our new design of cGA, we did some simulations on a Quartus II of Altera. For simulations, we use a Cyclone II EP2C70F896C6 with a clock of 50 MHz obtaining the results shown in Table 1. For the simulations we designed the cGA to solve the max-one problem, which consists on maximizing the number of ones of a bitstring, this implies a simple objective function, which is not costly. Table 1 shows a comparison between our design (cGA with CAPRNG) and an older version (cGA with RNG of 32 bits).

Table 1. Comparison of cGA designs for max-one problem.

Max-One		cGA with RNG of 32 bits	cGA with CAPRNG
8-bits	Runtime:	40.28μs	41.86μs
	Iterations:	335	348
	Combinational functions:	18796/68416	496/68416
	Dedicated logic registers:	826/68416	226/68416
	Embedded Multiplier 9-bits elements:	192/300	0/300
12-bits	Runtime:	41.48μs	43.06μs
	Iterations:	345	358
	Combinational functions:	28149/68416	703/68416
	Dedicated logic registers:	1214/68416	314/68416
	Embedded Multiplier 9-bits elements:	288/300	0/300

According to Table 1, although our design is ligthly slower for converge, the reduction of FPGA's resource consumption is impressive, while the cGA with RNG of 32 bits occupies approximately 10% of the FPGA's resources, our cGA with CAPRNG occupies about the 1% of FPGA's resources. This reduction of resource consumption is because we used a very simple *RNG* component. The Table 2 shows the differences of both components, demonstrating that CAPRNG is cheaper in the resource consumption of FPGA. Regarding to the number of iterations, our cGA executed more iterations because it emulates a population of 62 individuals (2^n-2), and the other design emulates a population of 50 individuals.

Table 2. Comparison of *RNG* components

Component		Data Synthesis
Random Number Generator based on Park & Miller Rules	Combinatorial functions:	1108/68416
	Dedicated logic registers:	826/68416
	Embedded Multiplier 9-bits elements:	192/300
Random Number Generator based on Cellular Automata	Combinatorial functions:	24/68416
	Dedicated logic registers:	6/68416
	Embedded Multiplier 9-bits elements:	0/300

6 Conclusions and Future Work

We obtained great benefits when substituted the RNG component based in Park and Miller Rules by the version based on Cellular Automata. We reduced about of 10 times the FPGA's resource consumption allowing a simpler and more parallel design (for RNG component case). This reduction of resource consumption will allow to implement another cGA variants in a better manner (for example, with elitism, mutation, etc.) or versions with more complex objective functions (because there will be more available resources for such purposes).

The convergence speed continues being attractive for applications that require to perform optimization in real time, although the objective function evaluation could consume more resources and computer time. According to the designer's abilities, it could be possible to parallelize, total or partially this section allowing a bigger convergence speed of the algorithm.

As a future work, we will conduct some experiments to design other cGA versions, as well as some simulations with objective functions capable of evaluating floating point values.

7 Acknowledgment

The authors thank the support of the Mexican Government (CONACYT, SNI, SIP-IPN, COFAA-IPN, and PIFI-IPN), also we appreciate the support of Altera Corporation and Víctor Maruri for the donation of DE2-70 kit and Quartus II academic licenses.

References

1. F. Cupertino, E. Mininno, E. Lino and D. Naso, "Optimization of Position Control of Induction Motors using Compact Genetic Algorithms", IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics, pp. 55-60, 2006.
2. M. Pelikan, D. Goldberg and F. G. Lobo, "A survey of optimization by building and using probabilistic models", Technical Report 99018, Illinois University, Illinois, USA, 1999.
3. X. Zhang, C. Shi and F. Hui. "FPGA-based Genetic Algorithm Kernel Design", ICES 2007, LNCS 4684, Springer-Verlag, 2007.
4. S. Scott and A. Seth, "HGA: A hardware-based genetic algorithm," in Proceedings ACM/SIGDA 3rd Int. Symposium. Field-Programmable Gate Arrays, pp. 53–59, 1995.
5. C. Apomtewan and P. Chongstitvatana, "A Hardware Implementation of the Compact Genetic Algorithm" in Proceedings 2001 IEEE Congress Evolutionary Computation, Seoul, Korea, Vol. 1, pp. 624–629, 2001.
6. J. C. Gallagher et al., "A Family of Compact Genetic Algorithms for Intrinsic Evolvable Hardware" in Proceedings 2004 IEEE Transactions on Evolutionary Computation, Vol.8, Issue 2,pp. 111–125, 2004
7. Y. Jewajinda and P. Chongstitvatana, "FPGA Implementation of a Cellular Compact Genetic Algorithm", NASA/ESA Conference on Adaptive Hardware and Systems, Vol. 22 June 2008, pp. 385 – 390, 2008.
8. G. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm", IEEE Transactions on Evolutionary Computation, Vol. 3, Issue 4, pp. 287-297, Nov. 1999.
9. P. D. Hortensius, R. D. McLeod and H. C. Card, "Parallel Random Number Generation for VLSI Systems Using Cellular Automata", IEEE Transactions on Computers, Vol. 38, Issue 10, October 1989, pp. 1466-1473.
10. F. Pardo and J. A. Boluda, VHDL Lenguaje para síntesis y modelado de circuitos. Alfaomega, 2nd edition, 2004, México (In Spanish).
11. A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computation, Springer, First Edition, 2003.
12. G. Harik, "Linkage Learning via Probabilistic Modeling in the ECGA", IlliGAL Technical Report 99010, Illinois University, Illinois, USA, 1999.

Dynamic Tracking of Multi-Agent Behavior Patterns Based on Topological Graphs

Huberto Ayanegui§ and Fernando Ramos‡

§ Universidad Autonoma de Tlaxcala
Facultad de Ciencias Basicas, Ingenieria y Tecnologia
Calzada de Apizaquito s/n, 90350,
Apizaco, Tlaxcala, Mexico.
‡ Tecnologico de Monterrey Campus Cuernavaca
Autopista del Sol, Km. 104, Col. Real del Puente.
Xochitepec, Morelos, Mexico. 62790
{huberto.ayanegui, fernando.ramos}@itesm.mx

Abstract. Two relevant problems are associated with behaviors of agents that cooperate in common tasks. The first one deals with adequate representations to recognize behavior patterns at several levels of abstractions. The second one deals with the tracking of behavior patterns. Both problems are confronted with the problem of dealing with multiple interactions submitted to dynamic conditions. This paper addresses the second problem through the construction of structures represented by topological graphs and their skeletons. Thus, behaviors of agents rely on structures formed by the relations between them. The topological graph is built by considering neighborhood relations between agents, where agents are represented by nodes and neighborhood relations by arcs. The topological conditions assure a planar graph, which is built by triangular sub-graphs, thus a structure where all the nodes can be connected. The problem of tracking behaviors is addressed by considering that, when the topological condition is broken, then a structure has changed, consequently the behavior has also changed. The skeleton of the primer topological graph allows a more abstract representation of structures that serves to reinforce the tracking of behavior patterns.

Key words: Pattern recognition, robotic soccer, formations, dynamic behavior.

1 Introduction

Multi-agent systems are one of the sub-disciplines of artificial intelligence which was introduced for the purpose of defining the rules and principles for developing complex systems and provides a mechanism for cooperating the agents [10]. The agents participating in a cooperative task, within real time environments, should be able to act autonomously an as a part of a team.

Two relevant problems are associated with behaviors of team agents that cooperate in common tasks. The first one deals with the discovery of behaviors

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 145-155

patterns which is increasingly needed in a variety of tasks, as we develop more autonomous robots and general information processing agents. For example, in multi-agent environments, an agent may need to make decisions based on the behavior of the other agents. Automatic discovery of the team strategy is a challenging problem which implies to dispose of rich representations able to support tactical and strategic behaviors. Most of the research involved in the construction of multi-agent behavior models does not consider relevant aspects of the team. Nevertheless, relevant aspects associated with any team should be taken into account in order to model its behaviors. These aspects include individual aspects, tactical aspects involving at least two agents; strategic aspects involving the whole team. To dispose of an expressive representation model which takes into account different aspects exhibited in a team of agents is key to discover behavior patterns in a complex domain[8].

The second problem deals with the tracking of the behaviors already discovered. Such tracking task becomes very complex because the dynamic conditions of the game brings about dramatic changes of positions and multiple interactions between players, which difficult the construction of models capable of discovering behaviors of teams playing soccer matches[6]. Although multiple frameworks have been developed for single agent plan recognition, there has been less work on extending these frameworks to multi-agent scenarios. We propose a method for tracking the dynamically changed structures of groups of agents through the construction of structures represented by topological graphs (TG) based on triangular planar graphs and their skeletons (STG).

The TG is built considering neighborhood relations between agents, where agents represent nodes and edges neighborhood relations. Thus, the relations construct a structure that can be represented by a graph G. The topological conditions assure a planar graph, which is built by triangular sub-graphs, thus a structure where all the nodes can be connected. The skeleton of the primer topological graph (STG) allows a more abstract representation of relations. The problem of tracking behaviors is addressed by considering that, when the topological condition is broken, then a structure has changed, consequently the behavior has also changed.

We situate our work in the domain of robotic soccer, in which it is potentially advantageous to be able to discover and track the multi-agent behaviors. Robotic soccer has been developed in simulation and with real robots[4]. This work deals with soccer-simulation category, where the teams are composed of 11 players, such as in real matches, evolving within a terrain whose area is a scale of the original soccer-terrains. Important aspects of multi-agent modeling have been developed in simulation[12]. The real robots are physical robotic agent computer controlled either on-board or off-board through radio communication. In both simulation and real robots, the agents must be fully autonomous, as human operators are not allowed to interfere once the game has started. In general, robotic or simulation agents observe the state of the environment, including the position of the other robots, and select actions based on the observed state.

The paper is organized as follows. Section 2 discusses the related work. Section 3 presents how the agents are organized, particularly in domains of soccer-agents. Section 4 describes the topological graphs and their skeletons used to track behavior patterns. Section 5 deals with the method for tracking formations and tactical behavior patterns using topological graphs and their skeletons. The conclusions and future works are presented in Section 6.

2 Related Work

Although there has been considerable research on the problem of single-agent behavior recognition, there has been substantially less work on multi-agent behavior recognition. Most of the previous work makes one of two assumptions: (1) each agent is a decoupled entity that can be analyzed individually using a single-agent activity algorithm or (2) the agents are always working together and can be analyzed as a single cohesive entity represented by a high-dimensional feature vector [11]. In either case, team structure is generally assumed to be static; this paper specifically addresses the problem of behavior recognition for teams with dynamic team structure.

Devaney and Ram [3] have worked on identifying behaviors in the sort of two dimensional spatial environment. In particular they looked at the movement of military troops during battle to identify behaviors through a combination of object tracking and pattern recognition. However, they are more focused on the problem of identifying particular repeated patterns of movement among the large amount of given movement data rather than trying to identify a particular model which captures all of the agents' movements.

Bezek and Bratko[2] present a method to discover pass patterns incorporating domain knowledge and providing a graphic representation for detected strategies. This work is focused on the increase of human comprehension. Although their approach obtains tactical behavior patterns, they only consider the players involved in the passes without taking into account the notion of team behaviors related with formations. Visser and colleagues recognized the formation of the opponent team using a neural networks model[13]. In the Visser's work observed player positions were the input of a neural network. The output was a predefined set of formations. If a classification can be done, the appropriate counter-formation is looked up and communicated to the players. The main difference with our approach is that Visser and colleagues did not represent relations between players. As Visser mentioned in his work, his approach is unable of tracking the changes of formations. This is because the lack of structures due to the absence of relations between players.

Riley and Veloso use a set of predefined movement models and compare these with the actual movement of the players in set play situation[9]. In new set play situations the coach then uses the gathered information to predict the opponent agent's behavior and to generate a plan for his own players. The approach uses probabilistic models and can be used in both off-line and on-line mode. The main drawback of Riley's model is that it is built based on individual movement

of players without taking into account the relationships between agents. Raines and colleagues presented a system called ISAAC[7] which analyzes a game in mode off-line using a decision tree algorithm to generate rules about the success of players. ISAAC uses the individual, relational and global models in an independent way. It tries to discover patterns in each level based on 'key events'. Key events are events that affect directly the result of the game, for instance, a kick to goal. Some of the key differences between ISAAC and the work presented in this paper are: we build a model of a team based on behavior patterns, independently of success or failures events. The global behavior discovered by ISAAC is not based on the formation of a team but in a general performance of the team in a set of games played previously. Thus, ISAAC is unable to discover the strategic behavior of a team. ISAAC doesn't take into account variants of behavior patterns.

3 Organization of multi-agent systems

Von Bertalanffy [14] gives the meaning of the somewhat mystical expression: The whole is more than the sum of its parts, is simply that constitutive characteristics are not explainable from the characteristics of isolated parts. Given the total of parts contained in an organization and the relations between them, the behavior of the system is derived from the behavior of the parts. So, a cooperative multi-agent system (CMAS) is an organization focused on how a loosely-coupled network of problem solvers can work together to solve problems that are beyond their individual capabilities.

Two relevant elements to consider in CMAS:

- *Functional analysis,* which describes the functions of a multi-agent system in its different dimensions.
- *Structural analysis,* which distinguishes between the various possible forms of organization and identifies some essential structural parameters.

The functional analysis leads to the identification of the principal functions that the components should fulfill. In the context of such an analysis, a CMAS can be seen as a system of roles. It is possible to make this analysis more precise developing these functions in several levels corresponding to the different points of view from which an organization can be studied. For example, in the soccer domain, at individual level, the role of players can be as goal- keeper, left defender, right mid-fielder or left forward. At zone level, all defenders (left, central and right) form the defensive zone, all mid-fielders form the middle zone and forwards form the attack zone. At team level, the soccer team can be adopt formations determined by offensive or defensive strategies.

The structural analysis attempts to bring order to all possible interactions between agents by isolating the abstract relationships which connect them and the way in which they evolve over time. The relations between soccer-agents occur through the multiple interactions taking place between them at each instant

of the game. For instance, an agent **A** can be related with the agent **B** because they interact through the exchange of passes to accomplish tactical plays. Another example of interactions occurs when several agents perform together a tactical play. In this situation, some of the relevant relations are determined by the structure that represents the distribution of soccer-agents accompanying the play. A formation is a structure based on positions of soccer-agents and relevant relations between them. For example, a code 5:2:3 represents a formation composed by five defenders, two mid-fielders, and three forwards. Goalkeepers are not counted because they are the one position forever. In general, strategies should be supported by formations in order to assure order, discipline and organization during a match [5].

3.1 Topological and Skeletons Graphs

Topological Graphs. A graph G is planar if it can be represented on a plane in such a way that the nodes represent different points and two edges should be encountered only at their ends. The intersection of two edges out of their ends breaks the planar property of the graph G. This graph G is also named as planar topological graph [1]. Two or more graphs are topologically the same if they can be transformed by elastic deformations until their form coincides, as shown in Figure 1.

The tactical plays are submitted to formations, in such a way that if a formation change or seems having changed, tactical plays are affected. Then, topological structures should guarantee that changes in formations have taken place if and only if the planar properties of the graph, representing structures, have been broken. Otherwise, formations have not changed.

This work uses a topological structure model based on relevant relations. The relevant relations used to build the topological structure are related with the notion of neighborhood. This way of doing supports well the idea of building formations, which is a key concept in this work. Thus, an agent remains related with his closer neighbor belonging to his zone (defensive (D), medium (M) or attack (A)), and his closer neighbor belonging to the neighbor zone as illustrated in Figure 2(a) and Figure 2(b). Figure 2(c) shows the integration of both kinds of relations for a 5:2:3 formation.

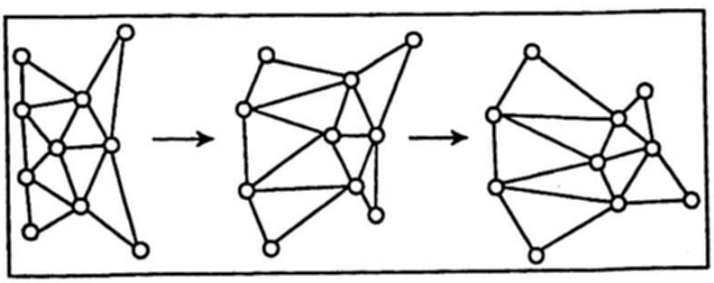


Fig. 1. Three representations of the same topological graph

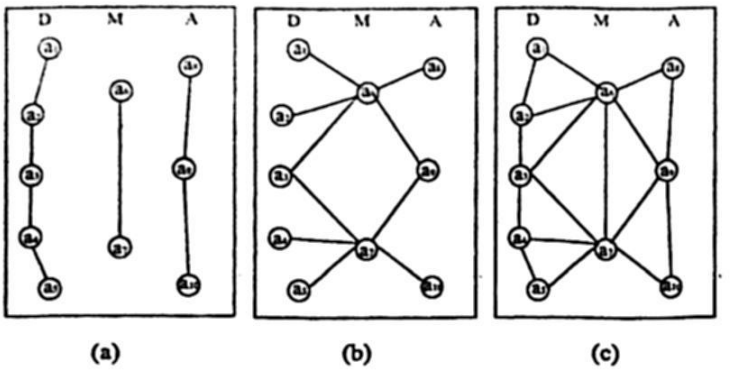


Fig. 2. (a) Neighbor nodes of the same zone are linked. (b) Neighbor nodes for neighbor zones are linked. (c) Topological graph including both kind of links.

The Skeleton Graphs. Some of the most important team behaviors are related with strategic and tactical plays [8]. As discussed above, the structure of a team serve to track the players involved in tactical plays at each instant of the match. That is, tactical plays occur, most of the time, under the context of formations. The discovery of tactical or team behaviors needs the tracking of both the positions of players at any instant of the game and relevant relations able to represent particular interactions between players. Nevertheless, the dynamic nature of soccer matches along with the multiple interactions between players difficult enormously the task of pattern recognition. On the one hand, the model based on topological graphs contributes importantly to manage the difficulties due to the dynamic nature of the soccer game to enable the tracking of formations. On the other hand, the skeleton of a topological graph serves in two ways: 1) to reinforce the break of topological conditions, because the skeleton suffers also of the break of topological conditions; 2) to facilitate the discovering of tactical plays with important information concerning the sub-structure participating in such plays.

4 Tracking of Formations and Tactical Plays

4.1 Tracking of Formations

Tracking of formations is important because a change of formation determine a potential changes of strategies and tactics.

The following example shows the break of topological conditions both for the *TG* and its *STG*, see Figure 3.

Frame 1 in Figure 3 shows a formation 4:3:3. Frame 2 is topologically similar to Frame 2 with the same formation 4:3:3. In Frame 3, player 7 moves down and player 8 moves right until almost it is aligned with forward players. However, we can verify that topological conditions are not yet broken. Frame 4 shows that the continuous movements of player 7 and player 8 have provoked the break of topological conditions of *TG*, because some intersections of arcs occur in

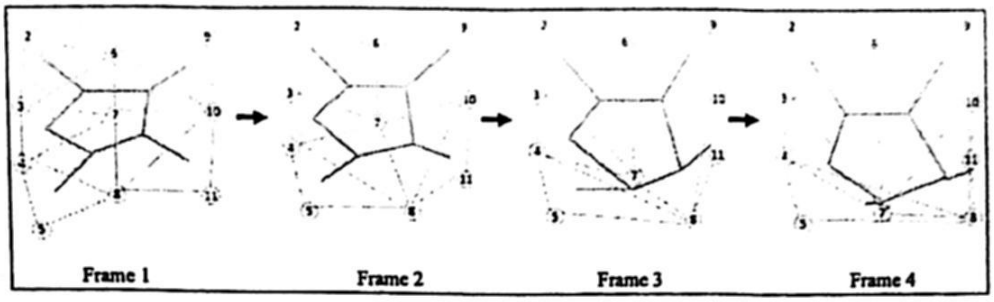


Fig. 3. Frame 4 shows the break of topological conditions for *TG* and *STG*.

the triangle formed by nodes 7,4,8. Given that, the *STG* is formed by linking neighbor nodes representing triangular graphs, the triangle formed by nodes 4,7,8 has not neighbor triangles well defined because the arc 4,8 is intersected by two other arcs. Thus, its topological conditions of the *STG* have been also broken. The construction of another *TG* and its *STG* is needed to represent the new formation, which is now a 4:2:4. This new formation corresponds to a more offensive strategy. We can verify that *STG* has reinforced the break of topological conditions, thus confirming a change of formation.

4.2 Tracking of Tactical Plays

Figure 4 shows a set of ball paths representing approaches, derived from offensive tactics, of the team located to the left to the opposite goal located at the right of the soccer terrain.

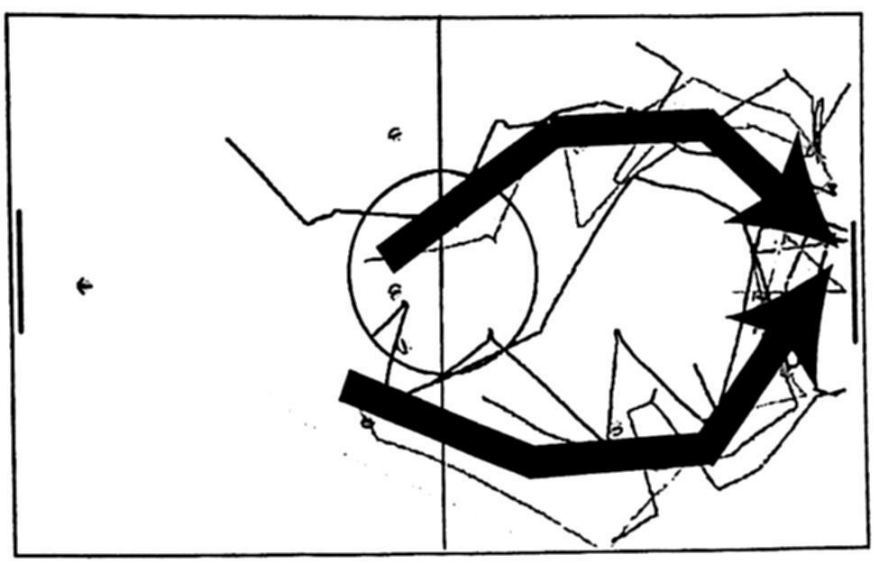


Fig. 4. Generalized paths obtained by applying the algorithm described in [8]. Generalized paths are drawn by thicker lines with an arrow at the end segment.

The details of the algorithm used to discover the tactical behaviors behind these ball paths are described in detail in [8]. Using freeman codes and generalization mechanisms, generalized paths were obtained from the set of paths illustrated in Figure 4. These generalized paths are represented by thicker lines with an arrow at the end segment of the path. This algorithm was not able to track the tactical plays and determine the structure of players associated with it. It is shown in the present work the method to track tactical plays and how the structure of players associated with it is determined. We argue that a tactical play relies on an adequate formation. Therefore, if the formation is represented by a structure, the tactical play should be also represented by a substructure, which is part of the formation structure. The substructure associated with the tactical play supports its performance.

In order to illustrate how the method to determine the structure associated with tactical plays works, let's consider the sequence of frames in Figure 5, which can be described as follows: ((frame1, cycle 1325), (frame2, cycle 1358), (frame3, cycle 1378), (frame4, cycle 1402), (frame5, cycle 1410), (frame 6, cycle 1418)). From the frame1 to frame5, there are a total of 93 frames, to complete the tactical play. Because of lack of space we do not show the 93 frames, but the most significant from the point of view of relevant moments of the tactical play. The sequence is extracted from a real match of the 2002 RoboCup World Championship. This sequence is one of the instances used to obtain the generalized paths shown in Figure 4.

These generalized paths do not provide neither information about players involved in the tactical play and nor a structure supporting the execution of the tactical play. The use of the *TG* and the *STG* cope with this problem. The fact of building the relations based on the concept of agents neighborhood relies on the fact that players pass the ball, most of the time, to a neighbor player and on the fact that semantically a more solid structure is made between neighbors of zones. Thus, the relations have two fold purposes: to build coherent structures representing correctly formations; to explicit relations that represent and support potential passes between players which could be linked to become tactical plays. Thus, recover a sequence of links (arcs) of the *TG* is a quite simple task. Meanwhile, *STG*'s serve as reference to rebuild a structure, or reconnect a not connected graph, able to support a well performance of the tactical play.

The sequence of frames illustrated in Figure 5 can be described as follows: (frame1, player 5 $\rightarrow$ player 6) $\prec$ (frame2, player 6 $\rightarrow$ player 9) $\prec$ (frame4, player 9 $\rightarrow$ player 10) $\prec$ (frame5, player 10 $\rightarrow$ player 11) $\rightarrow$ (frame6, player 11 $\rightarrow$ goal). The character $\prec$ means that a frame X is before than frame Y. The arrow $\rightarrow$ means that a player X pass the ball to player Y. So, the play has been executed by following the sequence of passes: 5$\rightarrow$6, 6$\rightarrow$9, 9$\rightarrow$10, 10$\rightarrow$11, 11$\rightarrow$goal. If we link the passes, we obtain a path quite similar to the corresponding generalized path. The players executing the passes were extracted thanks to the *TG*. The arcs representing passes are arrowed showing the direction of passes. Thus, the substructures participating in the play will be the triangles associated with arrowed arcs. Figure 6(a) shows such substructure.

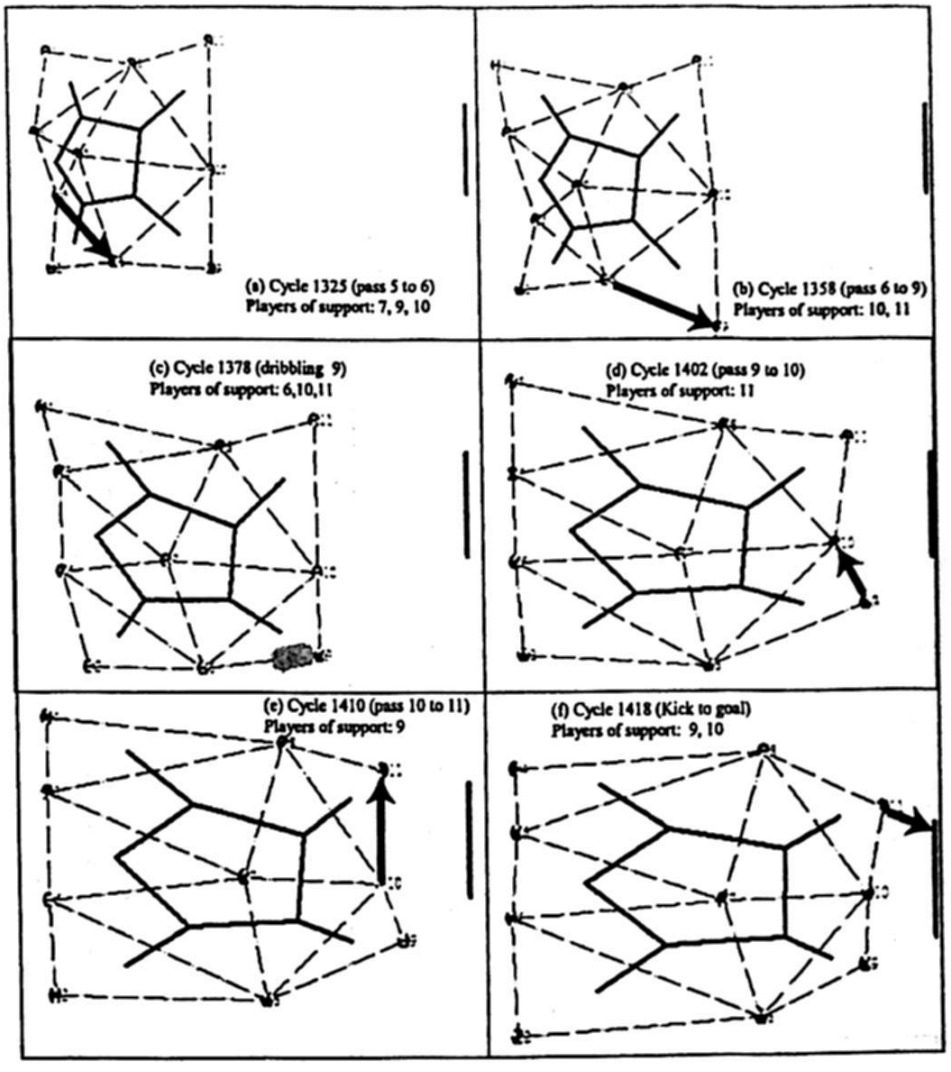


Fig. 5. Sequence of frames representing a tactical play

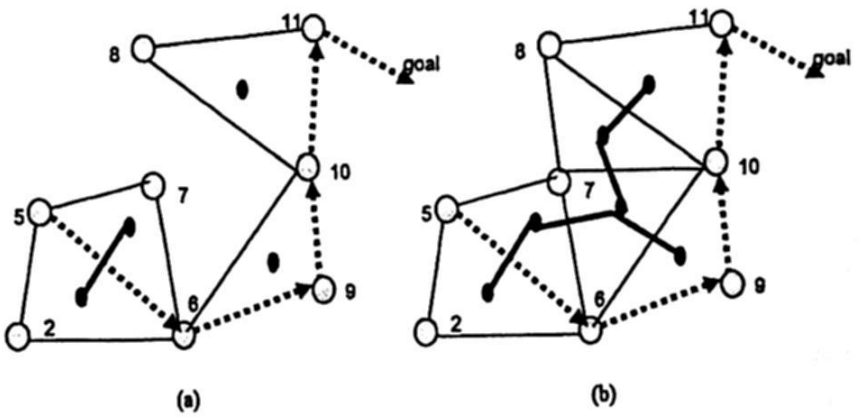


Fig. 6. (a)*STG* not connected. (b)*STG* reconnected.

Based on Figure 6(a) above, the topological property of the *STG* has been broken, which can be interpreted that the play has not a connected support, thus the tactical play are not robust and if one of the players fails, the tactical play has an important risk of not being achieved. The *STG* serves as reference to convert the not connected graph into a connected one. Then, it results the subgraph illustrated by Figure 6(b), which represents the tactical play associated with players serving as support to improve the robustness of the play. For instance, if the tactical play takes place based on the graph of Figure 6(a), in case of an eventual fail of player number 9 or 10 the play is not supported by other players that could recover the failed tactic. Instead, the structure rebuilt once the part of the skeleton is added, player number 7 is incorporated automatically who can support the performance of the tactic play. Thanks to the operation of insertion of skeleton's links, the tactical play has been better characterized.

5 Conclusions

The discovery of tactical plays and formations supporting strategies of team represents relevant information to implement counter strategies or tactics to reduce the performance of the opposite team or, in the best of cases, to beat it.

Nevertheless, the dynamic nature of soccer matches, along with the multiple interactions between players, difficult enormously the task of discovery. The model based on topological graphs and their skeletons have contributed importantly to manage the difficulties for tracking behavior patterns as they evolve within dynamic environments such as soccer game. A relevant contribution in this work is the use of skeletons, which are also topological graphs. On the one side, skeletons can reinforce the break of topological conditions and confirm the change of formations. On the other side, skeletons serve to track tactical plays, with important information concerning the players participating in such plays and the sub-structure supporting them.

As future work the discovery of defensive tactics is necessary to have a richer spectrum of the tactics used in a match.

References

1. Claude Berge. *Graphes.* Guathier-Villars, 1983.
2. Andraz Bezek, Matjaz Gams, and Ivan Bratko. Multi-agent strategic modeling in a robotic soccer domain. In Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone, editors, *AAMAS*, pages 457–464. ACM, 2006.
3. Mark Devaney and Ashwin Ram. Needles in a haystack: Plan recognition in large spatial domains involving multiple agents. In *AAAI/IAAI*, pages 942–947, 1998.
4. Hiroaki Kitano, Milind Tambe, Peter Stone, Manuela M. Veloso, Silvia Coradeschi, Eiichi Osawa, Hitoshi Matsubara, Itsuki Noda, and Minoru Asada. The robocup synthetic agent challenge 97. In Hiroaki Kitano, editor, *RoboCup*, volume 1395 of *Lecture Notes in Computer Science*, pages 62–73. Springer, 1997.

5. Gregory Kuhlmann, Peter Stone, and Justin Lallinger. The ut austin villa 2003 champion simulator coach: A machine learning approach. In Daniele Nardi, Martin Riedmiller, Claude Sammut, and Jos Santos-Victor, editors, *RobuCup*, volume 3276 of *Lecture Notes in Computer Science*, pages 636–644. Springer, 2004.
6. Andreas D. Lattner, Andrea Miene, Ubbo Visser, and Otthein Herzog. Sequential pattern mining for situation and behavior prediction in simulated robotic soccer. In Ansgar Bredenfeld, Adam Jacoff, Itsuki Noda, and Yasutake Takahashi, editors, *RoboCup*, volume 4020 of *Lecture Notes in Computer Science*, pages 118–129. Springer, 2005.
7. Ranjit Nair, Milind Tambe, Stacy Marsella, and Taylor Raines. Automated assistants for analyzing team behaviors. *Autonomous Agents and Multi-Agent Systems*, 8(1):69–111, 2004.
8. Fernando Ramos and Huberto Ayanegui. Discovering tactical behavior patterns supported by topological structures in soccer agent domains. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 1421–1424, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.
9. Patrick Riley, Manuela Veloso, and Gal Kaminka. An empirical study of coaching. In H. Asama, T. Arai, T. Fukuda, and T. Hasegawa, editors, *Distributed Autonomous Robotic Systems 5*, pages 215–224. Springer-Verlag, 2002.
10. Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 2003.
11. Gita Sukthankar and Katia Sycara. Simultaneous team assignment and behavior recognition from spatio-temporal agent traces. In *AAAI*. AAAI Press, 2006.
12. Milind Tambe. Agent architectures for flexible, practical teamwork. In *In Proceedings of the National Conference on Artificial Intelligence*, pages 22–28, 1997.
13. Ubbo Visser, Christian Drcker, Sebastian Hbner, Esko Schmidt, and Hans-Georg Weland. Recognizing formations in opponent teams. In Peter Stone, Tucker R. Balch, and Gerhard K. Kraetzschmar, editors, *RoboCup*, volume 2019 of *Lecture Notes in Computer Science*, pages 391–396. Springer, 2000.
14. Ludwig Von Bertalanffy. *General System Theory: Foundations, Development, Applications*. Georges Braziller, Inc., New York, USA, 1963.

Better Interpretation of Numerical Data Sets by Relative and Absolute Typicality of Fuzzy Clustering Algorithms

B. Ojeda-Magaña[1], R. Ruelas[1], M.A Corona-Nakamura[2], and D. Andina[3]

[1] Departamento de Ingeniería de Proyectos-CUCEI, Universidad de Guadalajara.
José Guadalupe Zuno No. 48, C.P. 45101 Zapopan, Jalisco, México.
benojed@hotmail.com, rruelas@newton.dip.udg.mx.
[2] Departamento de Ciencias Computacionales-CUCEI, Universidad de Guadalajara.
Revolución 1500. C.P. 44840 Guadalajara, Jalisco, México.
[3] Group for Automation in Signals and Communications (GASC)
Universidad Politécnica de Madrid.

Abstract. In this work we take the concept of typicality from the cognitive and psychological point of view, and we apply their meaning to the interpretation of numerical data through fuzzy clustering algorithms, looking for a better profit of the typicality in the clusters found, and a better interpretability of the processed data. With the Fuzzy c-Means clustering algorithm (FCM) we get a relative typicality (membership degree), and with the Possibilitistic c-Means (PCM) an absolute typicality (typicality value). Nowadays, some hybrid clustering algorithms have been developed that combine both algorithms and where both typicalities are available; one such algorithm is the Possibilistic Fuzzy c-Means (PFCM), which is the basis of the algorithm used in this work. Thus, it is important to make better use of both typicalities, for learning and analysis of data. To get some results we use a synthetic data set. The results clearly show the advantages of the information obtained about the data set, taking into account the different meaning of typicalities and the availability of both values with the clustering algorithm used.

Keywords: Clustering, Relative Typicality, Absolute Typicality.

1 Introduction

The objective of clustering algorithms is to find an internal structure in a numerical data set into n different subgroups, where the members of each subgroup have a high similarity with its prototype (centroid, cluster center, signature, template, code vector) and a high dissimilarity with the prototypes of the other subgroups. This justifies the existence of each one of the subgroups.

The clustering algorithms help us to get a simplified representation of a numerical data set into n subgroups, as a way to get a better comprehension and knowledge about data. Also, the clustering algorithms that partition in a hard, fuzzy, probabilistic or possibilistic way a given space, according to a data set and

Special Issue in Advances in Pattern Recognition
Research in Computing Science 44, 2009, pp. 157-166

after a learning process, provide a set of prototypes as the most representative elements of each subgroups.

In this work we propose to take the interpretation of the typicality concept according to the cognitive and psychological point of view, such that it is possible a better interpretation of data. Rosch *et al* [1] proposed a theory of prototypes for the classification of objects that belong to a semantic characteristic, taking into account their proximity to a prototype, according to a given criterion. So, in each category there must be an internal resemblance among category members, and an external dissimilarity meaning that the similarity with the members of the other categories is low.

A prototype is based on the typicality notion; all members belonging to the same category do not represent it in the same way, that is, some members are more typical than others. Berlin *et al* [2] were among those who first recognized this variation inside the categories, when they discovered the focal color which is a psychological definition for color identification. They show a similarity among different languages in color categorization, the so-called "basic colors", that is, each color is represented by the best example (some nuances of colors are more representative than others), and inside a category of color there is no other color. For example, the scarlet color is not a basic color because it is in the red color category. Thus, the colors are not uniform and they are represented by a center or better example and peripheral members. However, if we are interested knowing the edge of each category of color, such as the threshold where the color stops being red but begins to be orange, for example, we can use fuzzy sets. Zadeh [3] has proposed the theory of fuzzy sets in order to solve the problem of boundaries. With fuzzy sets, the members of a particular category have a membership degree in the [0, 1] interval, where 1 is assigned to the most representative members, and 0 to the elements that are not members of the category.

In this work we use the concepts of typicality and membership degrees in order to categorize linguistic concepts, looking for a better understanding of the information extracted from a numerical data set through the Possibilistic Fuzzy c-Means (PFCM) clustering algorithm.

This paper is organized as follows: Section 2 presents the typicality and vagueness concepts. Section 3 contains the relative and absolute typicalities, as well as the PFCM clustering algorithm. Section 4 presents the concepts of typicality applied to a synthetic numerical data set. Finally, Section 5 presents the main conclusions.

2 Typicality and vagueness

In the search of prototypes Hampton [4] mentions the phenomena of vagueness, typicality, genericity and opacity. In this work we use vagueness, and typicality, both different but of interest as they help us better define the characteristics of the concepts. The typicality alludes to a measure in which the objects under study are considered good examples of the concept [4]. For example, in the category of birds, dove is a typical case as it has the following characteristics: it

can fly, it has feathers, it puts eggs, and it has a nest in a tree. On the other side, an atypical case is the penguin because it satisfies only some characteristics, but not all. For this example, the prototype has the most common characteristics or the mean values of these characteristics.

The categories of some concepts can be vague or fuzzy, i.e. there exist objects whose membership to the category is uncertain, and this is not due to a lack of knowledge, but to the lack of a clear rule defining the edges of the categories [5]. Some classical examples are the adjectives: *high*, or *red*; or the nouns: *vegetable*, or *chair*. Vagueness is mostly a question of truth (yes or not), and it represents a measure of correspondence of an object with a conceptual category. For some categories the edges are defined in an easier way, such as the category of birds. However, for other categories, as for the adjective *high*, the edges are not so easy to define, and therefore a membership degree in [0, 1] is used.

For a better understanding of the differences between typicality and vagueness, take an example of Osherson *et al* [4], in order to appreciate the difference between typicality and vagueness, they note that *... many people believe that penguins are an atypical case of birds, only a few doubt that they are birds in reality; in this case the typicality is involved, but not the vagueness.*

The membership degree of the penguin and the dove to the category of birds is 1. However, the dove is more typical than the penguin. For the concept of *high* there is no example with maximum typicality, due to the fact that in some contexts height can be increased infinitely. On the other hand, the membership degrees of the concept *high* represents the variation on the certainty degrees using values in 0 and 1, both included, and they provide the edge to determine at which height something is "*high*" and at which height it is not.

3 Relative and Absolute Typicality in the Clustering Algorithms

The partition clustering algorithms have a great similarity to the theory of prototypes, although the latter is related to building categories about concepts and the clustering algorithms focus on the classification of numerical data; however both approaches have the same objective.

In this section we give a better interpretation of the typicality of the fuzzy clustering algorithms, based on a psychological and cognitive interpretation, as presented in the previous section, as a way to gain a greater knowledge than usual from numerical data sets.

3.1 Fuzzy c-Means Clustering Algorithm (Relative Typicality)

The first to use fuzzy sets for clustering was Ruspini [6]. After that, Dunn [7] proposed the first fuzzy clustering algorithm named Fuzzy c-Means (FCM) with the parameter of fuzziness m equal to 2. Later on Bezdek [8] generalized this algorithm. The FCM is an algorithm where the membership degree of each point to each fuzzy set A_i is calculated according to its prototype. The sum of all the

membership degrees of each individual point must be equal to one. Therefore the degree of membership to a particular fuzzy set is influenced by the position of all the prototypes of the fuzzy sets, and that is the reason why Pal *et al* [9] interpret the membership as a relative typicality.

With the FCM, the calculus of the membership degree of a point z_k to the fuzzy sets A_i is inversely proportional to the relative distance of this point to the prototypes (centers) of the fuzzy sets. Pal *et al* [10] show a deficiency of the algorithm when there are several equidistant points from two prototypes, as the membership degrees to both fuzzy sets are the same, but the distance to the prototypes is different; one point is further than the other. These data must be handled with care as they do not represent both prototypes in the same way. Another disadvantage of the FCM algorithm is its sensitivity to noise, or points far away from a concentration of prototypes.

3.2 Possibilistic c-Means Clustering Algorithm (Absolute Typicality)

The Possibilistic c-Means (PCM) clustering algorithm was proposed by Krishnapuram and Keller [11], and its principal characteristic is the relaxation of the restriction that gives the relative typicality property of the FCM. As a consequence, the PCM help us to calculate a similarity degree between data points and each one of the prototypes; value known as absolute typicality or simply typicality [9]. The nearest points to a prototype are identified as typical, whereas the furthest points as atypical, and noise if their typicality is zero or almost zero [12]. The PCM is very sensitive to the initial value of its parameters. Also, to avoid the coincidence of several prototypes, it is convenient to use the modified objective function proposed by Tim *et al* [13–15], which contains a restriction resulting in a repulsion of the prototypes and avoiding prototypes located at the same place.

3.3 Categories, Typicality and Clustering Algorithms

The prototypes are selected as the best examples, according to a given criterion, to represent groups, and they have the most important characteristics. In the case of birds, for example, the dove is more typical than the ostrich and the penguin, because it has more characteristics of a bird. However, ostriches and penguins are members of the category of birds. Therefore, there is an internal resemblance among the members of a group, and an external dissimilarity to the members of other categories, even when several categories share some characteristics, as it happens with birds and reptiles, as both kinds of animals reproduce by eggs. However, each group has its own characteristics that define them as members of a particular group, and different to the others.

A similar situation happens with a numerical data set, that is, it is possible to take into account an external dissimilarity and an internal resemblance.

- *External dissimilarity*: it results from fuzzy clustering algorithms, because the membership degree of a data point to a group depends on the membership degrees to the rest of groups. Finally, the data point is considered a member of the group to which it has the maximum membership degree. In other words, it belongs to the nearest group.

- *Internal resemblance*: it results from algorithms such as the PCM, and represents the resemblance between a data point and a prototype. For this reason it is possible to establish thresholds in order to identify typical, atypical and noisy data.

Thus, the FCM and the PCM algorithms provide information about a numerical data set, and in this way they help us to better understand its structure. Pal *et al* [16] have proposed a hybrid algorithm, the PFCM, that provides of the advantages of the FCM and PCM algorithms, and it avoids some problems of these algorithms used separately. This algorithm is described in the next Section.

3.4 Hybrid prototypes

Pal *et al.* [9] have proposed a hybrid algorithm that uses membership degrees (relative typicality), as well as typicality values (absolute typicality), trying to take profit of both algorithms in a single one. However, due to a problem with a restriction on the typicality values, they modified the algorithm and proposed the Possibilistic Fuzzy c-Means (PFCM) algorithm [16].

The PFCM algorithm is based on the Euclidian distance and the identified clusters are constrained to spherical shapes. Attempting to find a more general result, we decided to use the Gustafson Kessel Possibilistic Fuzzy c-Means (GKPFCM) algorithm proposed by the authors [17], which is based on the Mahalanobis distance according to the Gustafson and Kessel method [18], such that the identified groups are better adapted to the distribution of the data set.

`GKPFCM clustering algorithm`

I Initialize the prototypes (centers) $v_i, i = 1, .., c$. These are regularly obtained on a random basis.

II Find the value of the parameter δ_i. To do this we would run the FCM clustering algorithm [19] and then use Equation (1) proposed by Krishnapuram and Keller [11] [20]:

$$\delta_i = K \frac{\sum_{k=1}^{N} \mu_{ik}^m \|z_k - v_i\|_A^2}{\sum_{k=1}^{N} \mu_{ik}^m} \tag{1}$$

$K > 0$, the most common option is $K = 1$.

III Propose new values for parameters that are added to the algorithm PFCM, they are (ρ_i, β, γ), regularly the first two parameters remain constant $\rho_i = 1$, $i = 1, .., c$, and $\beta = 10^{15}$, the only parameter that is modified is the value of γ, and it can take values between 0 and 1.

IV Propose values of the parameters a, b, m, and η. These parameters play an important role in the calculation of the prototypes and the membership and typicality values.

V Calculate the covariance matrices for each group:

$$F_i = \frac{\sum_{k=1}^{N}(\mu_{ik})^m(z_k - v_i)(z_k - v_i)^T}{\sum_{k=1}^{N}(\mu_{ik})^m}, \quad 1 \le i \le c. \tag{2}$$

and estimate the covariance as proposed by Babuska:

$$F_i = (1-\gamma)F_i + \gamma \det(F_0)^{\frac{1}{n}} I \tag{3}$$

Extract the eigenvalues λ_{ij} and eigenvectors ϕ_{ij}, matrix F_i, and find all $\lambda_{i,max} = max_j \lambda_{ij}$ and $\lambda_{i,max} = \lambda_{ij}/\beta$, $\forall_j$ which satisfies $\lambda_{i,max}/\lambda_{i,j} \ge \beta$.

Finally F_i is rebuilt with the equation (4)

$$F_i = [\phi_{i,1}, \ldots, \phi_{i,n}] diag(\lambda_{i,1}, \ldots, \lambda_{i,n})[\phi_{i,1}, \ldots, \phi_{i,n}]^{-1}, \quad 1 \le i \le c \tag{4}$$

VI Calculation of distances:

$$D^2_{ikA_i} = (z_k - v_i)^T[\rho_i \det(F_i)^{-1/n} F_i^{-1}](z_k - v_i) \tag{5}$$

VII Determine the membership matrix $U = [\mu_{ik}]$ using equation (6)

$$\mu_{ik} = \left(\sum_{j=1}^{c} \left(\frac{D_{ikA_i}}{D_{jkA_i}} \right)^{2/(m-1)} \right)^{-1}, \quad 1 \le i \le c; \quad 1 \le k \le n \tag{6}$$

VIII Determine the typicality matrix $T = [t_{ik}]$ using equation (7)

$$t_{ik} = \frac{1}{1 + \left(\frac{b}{\gamma_i} D^2_{ik_A} \right)^{1/(\eta-1)}}, \quad 1 \le i \le c; \quad 1 \le k \le n \tag{7}$$

IX Modify the v_i prototypes according to equation (8).

$$v_i = \sum_{k=1}^{N}(a\mu_{ik}^m + bt_{ik}^\eta)z_k \Big/ \sum_{k=1}^{N}(a\mu_{ik}^m + bt_{ik}^\eta), \quad 1 \le i \le c. \tag{8}$$

X Verify that this error is within the proposal tolerance ε.

$$\|V_{nuevo} - V_{viejo}\|_{error} \le \varepsilon$$

XI If the error is greater than ε return to step V.

4 Application of the Typicality Concept to Numerical Data Sets

For this work we use a synthetic numerical data set [21]. For that, two Gaussian clouds with 400 points were generated. The covariance matrix is the same for both subsets. Forty noisy points were added at the corners of the subsets (see Fig. 1a).

$$\sum_1 = \sum_2 = \begin{bmatrix} 4.47 & 0 \\ 0 & 0.22 \end{bmatrix}, \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 2.5 \end{bmatrix}$$

In order to get a better knowledge from a data set, the following algorithm was proposed:

I Propose the parameters a, b, m, and η, and the number of c clusters before the execution of the GKPFCM algorithm.
II Run the GKPFCM algorithm to estimate the relative typicality provided by the U matrix, and the absolute typicality from the T matrix.
III From the U values, providing the external dissimilarity, find the boundaries among the clusters.
IV From the T values providing an internal dissimilarity, and using thresholds, it is possible to differentiate among typical, atypical, or noise.

To reduce the effects of noise in the prototype it is necessary to make a good choice of parameters a, b, m, and η for the GKPFCM algorithm. The parameters a and b have a great influence on the calculation of the prototype. Pal et al [16] recommended a value of b greater than the value of a, such that the prototypes are more influenced by the membership values. On the other hand, it is recommended a small value for η and a value greater than 1 for m. Nevertheless, a too high value of m reduces the effect of membership of data to the clusters, and the algorithm behaves as a simple PCM. For this work, we use the values $a = 1, b = 5, m = 2$, and $\eta = 2$ for the GKPFCM algorithm. The estimated values for the prototypes are: $\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0.1659 & -0.0112 \\ 0.2273 & 2.5721 \end{bmatrix}$ (see Fig. 1b).

The U matrix provides the external dissimilarity, and it helps to discriminate the clusters, that is, the maximum value of each vector indicates the cluster to which each data point belongs. In Fig. 2a we can see that the membership degree of noisy data is very high, as a consequence of the relation of data with the prototypes. In Fig. 3a we use the external dissimilarity to separate clusters with a straight line.

The T matrix provides the internal dissimilarity which is used for the partition of data, and to know the data within each cluster. In this case we are interested how near or far the data points are from the prototypes. Thus, in this work we propose a threshold such that typical and noise data can be differentiated. In Fig. 3b we use a threshold of 0.01 for dividing the typicality T matrix into two submatrices (T_1, T_2), T_1 for data with typicality values greater or equal

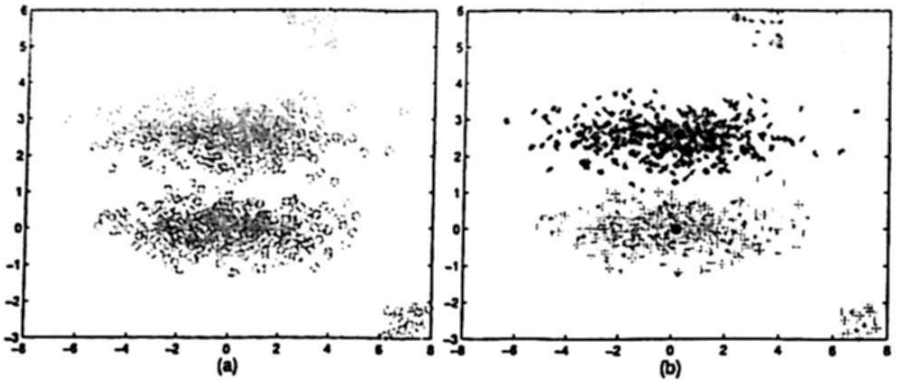


Fig. 1. (a) Synthetic data set. (b) Partition of the data set with the GKPFCM algorithm.

than the threshold, whereas T_2 for the other data, or data considered as noise as their typicality values are below the proposed threshold. Data of T_2 have a very low typicality values as they are far away from the prototype. In this paper, the threshold value has been proposed empirically. However, this must be a very low value as we try to identify data that really represent the corresponding cluster, and to eliminate those data considered as noise.

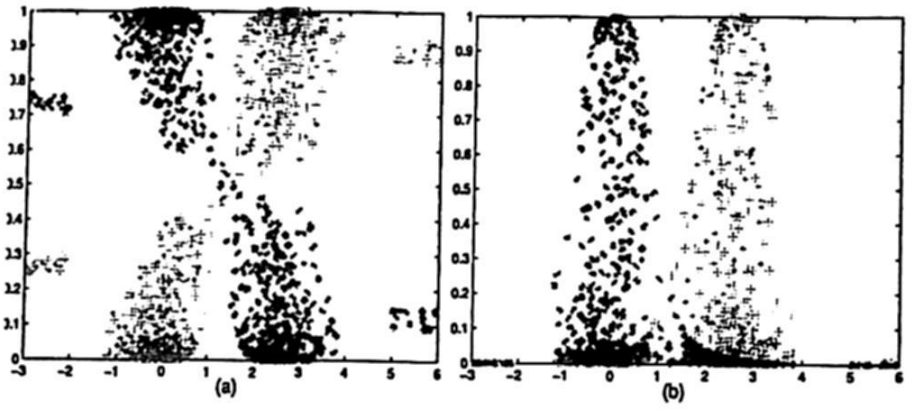


Fig. 2. ((a) Projection of the membership degrees of the U matrix, and (b) Projection of the typicality values of the T matrix.

5 Conclusions

Categorizing data into concepts based on the theory of prototypes, allows us to better understand the problem of classification and its application to a numerical data set. The external dissimilarity among data points of a numerical data set is evaluated with algorithms that use the relative typicality, and the internal

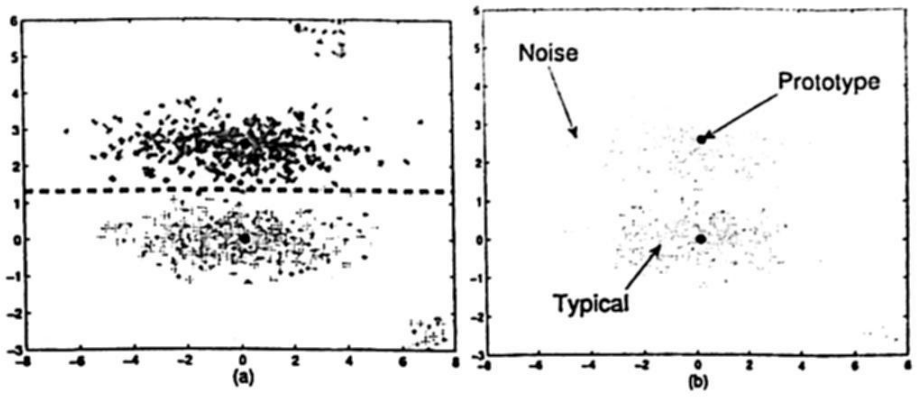


Fig. 3. Boundary generated from the external dissimilarity. (b) Separation of clusters using the internal dissimilarity (see prototypes, typical data, and noise).

resemblance is evaluated with algorithms that use absolute typicality. Using both typicalities through a single algorithm, as the GKPFCM, we can get more information about the internal structure of a numerical data set. In this work we have tried to relate the classification made by human beings and the automatic algorithms. That is the reason we have tried to meld the theory of prototypes and the partitional clustering algorithms.

References

1. Rosch E; and Mervis C. Family resemblance: Studies in the internal structure of categories. *Cognitive Psychology*, 7:573–605, 1975.
2. Berlin B; and Kay P. *Basic Color Terms: Their Universality and Evolution.* University of California Press, Berkley CA, 1968.
3. Zadeh L. A. Fuzzy sets. *Information and Control*, pages 338–353, 1965.
4. Osherson D; and Smith E. E. On typicality and vagueness. *Cognition*, 64:189–206, 1997.
5. Hampton J. A. Typicality, graded membership, and vaguess. *Cognition*, 31:355–386, 2007.
6. Ruspini E. Numerical method for fuzzz clustering. *Inf. Sci. 2*, pages pp. 319–350, 1970.
7. Dunn J. C. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, Vol 3:pp. 32–57, 1973.
8. Bezdek J. C. *Pattern Recognition With Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.
9. Pal N. R; Pal S. K; and Bezdek J. C. A mixed c-means clustering model. In *IEEE International Conference on Fuzzy Systems, Spain*, 11-21, 1997.
10. Pal N. R; Pal S. K; Keller J. M; and Bezdek J. C. A new hybrid c-means clustering model. In *Proc. of the IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE-04, I. Press, Ed.*, 2004.
11. Krishnapuram R; and Keller J. A possibilistic approach to clustering. *International Conference on Fuzzy Systems*, Vol. 1, No. 2:pp. 98–110, 1993.

12. Ojeda-Magaña B, Ruelas R, Buendía F.S, and Andina D. A greater knowledge extraction coded as fuzzy rules and based on the fuzzy and typicality degrees of the gkpfcm clustering algorithm. *Intelligent Automation and Soft Computing*, Vol 15, no. 4:555–557, 2009.
13. Timm H; Borgelt C; Döring C; and Kruse R. Fuzzy cluster analysis with cluster repulsion. In *presented at the Euro. Symp. Intelligent Technologies (EUNITE), Tenerife, Spain*, 2001.
14. Timm H and Kruse. A modification to improve possibilistic fuzzy cluster analysis. In *Conference Fuzzy Systems, FUZZ-IEEE, Honolulu, HI, USA*, 2002.
15. Timm H; Borgelt C; Döring C; and Kruse R. An extension to possibilistic fuzzy cluster analysis. *Fuzzy Sets and systems*, Vol 147, No 1:pp 3–16, 2004.
16. Pal N. R; Pal S. K; Keller J. M; and J. C Bezdek. A possibilitic fuzzy c-means clustering algorithm. *IEEE Transactions on Fuzzy Systems*, 13, no. 4:517–530, 2005.
17. Ojeda Magaña B; Ruelas R; M. A. Corona Nakamura; and Andina D. An inprovement to the possibilistic fuzzy c-means clustering algorithm. In *Image Procesing and Biomedicine*, TSI Press Series on Intelligent Automation and Soft Computing. Vol 20, pp 585-592, 2006.
18. Gustafson E. E and Kessel W. C. Fuzzy clustering with a fuzzy matriz de covarianza. *Proceedings of the IEEE CDS, San Diego CA*, pages 761–766, 1979.
19. Krishnapuram R; Bezdek J. C, Keller J and Pal N.R. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing.* Boston, London, 1999.
20. Krishnapuram R. and Keller J. The possibilistic c-means algorithm: Insights and recommendations. *International Conference on Fuzzy Systems*, Vol 4, No 3:pp 385–393, 1996.
21. Marie-Jeanne Lesot and Rudolf Kruse. Gustafson-kessel-like clustering algorithm based on typicality degrees. In *Information Processing and Management of Uncertainty in Knowledge-based Systems*, 2006.

Author Index
Índice de Autores

Editorial Board of the Volume

Comité Editorial del Volumen

Additional Reviewers

Árbitros adicionales

Diaz-Pernil, D.
Gago-Alonso, A.
Garcia, E.
Gil-Rodríguez, José L.
Hernández-Rodríguez, S.
Hernández-Rojas, L.
Hernandez-Sierra, G.
Olverá-López, José A.
Pernici, F.
Tscherepanow, M.
Sánchez Díaz Guillermo
Franco Arcega Anilu

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Noviembre de 2009.
Printing 500 / Edición 500 ejemplares.

The Mexican Workshop on Pattern Recognition (MWPR) is a forum for exchanging scientific results and experiences, as well as sharing new knowledge, and increasing the co-operation between research groups in pattern recognition and related areas, in Mexico. MWPR2009 was organized by the Computer Science Department of the National Institute of Astrophysics, Optics and Electronics (INAOE) and the Mexican Association for Computer Vision, Neural Computing and Robotics (MACVNR), and was sponsored by the Research and Advanced Studies Center campus Guadalajara (CINVESTAV-GDL), and the Center for Computing Research of the National Polytechnic Institute (CIC-IPN).

This volume presents 18 original peer-reviewed research papers on Pattern Recognition, Image and Document Analysis, and real word applications. The papers are structured into four thematic fields:

- Computer Vision
- Language and Document Processing
- Image Processing and Texture Analysis
- Pattern Recognition Techniques

ISSN: 1870-4069
www.ipn.mx
www.cic.ipn.mx

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

INSTITUTO POLITÉCNICO NACIONAL
"La Técnica al Servicio de la Patria"