

Advances in Pattern Recognition

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

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

Associate Editors:

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

Editorial Coordination:

María Fernanda Ríos Zacarías

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 112**, Junio 2016. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 112**, June 2016. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 112

Advances in Pattern Recognition

J. A. Olvera-López
V. Ayala-Ramírez
J. A. Carrasco-Ochoa
J. F. Martínez-Trinidad
X. Jiang (eds.)



Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2016

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2016

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.rcs.cic.ipn.mx>

<http://www.ipn.mx>

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

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

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

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

Preface

The 2016 Mexican Conference on Pattern Recognition (MCPR 2016, June 22-25) was jointly organized by the University of Guanajuato and the Computer Science Department of the National Institute for Astrophysics Optics and Electronics (INAOE); with the consent of the Mexican Association for Computer Vision, Neurocomputing and Robotics (MACVNR) and the International Association of Pattern Recognition (IAPR). MCPR 2016 was the eighth event in the series, MCPR series of conferences aim to provide a forum for the exchange of scientific results and new knowledge, as well as promoting cooperation among research groups in Pattern Recognition and related areas in Mexico and around the world.

In this year the fourth MCPR Postgraduate Students' Meeting (MCPR2016-PSM) allowed discussing research works from both Master and PhD students in order to receive feedback from experienced researchers and advices for future directions, as well as promoting their participation in conference events.

This RCS volume contains original contributions selected through a reviewing process by our Program Committee. These contributions are derived from either Master or PhD students' researches about Pattern Recognition and related areas. We cordially thank all authors for their submitted contributions to build this volume as well as the Program Committee for providing valuable comments and evaluations to authors.

We hope this RCS volume will be useful to the reader interested in Pattern Recognition and related areas, and hope that the meeting itself will provide a fruitful forum to enrich the collaboration between students and the broader Pattern Recognition community.

We would like to thank Research in Computing Science journal editors for giving us the opportunity of continuing to publish a special issue devoted to MCPR-PSM. Finally, but not less important, our thanks go to the University of Guanajuato for providing a key support to this event.

The submission, reviewing, and selection process was supported for free by the EasyChair system [www.easychair.org].

J. A. Olvera-López
V. Ayala-Ramírez
J. A. Carrasco-Ochoa
J. F. Martínez-Trinidad
X. Jiang
June 2016

Table of Contents

Page

2D Image Segmentation Through the One Dimensional Mumford-Shah Functional.....	9
<i>A. Lizbeth Cortés Cortes, Carlos Guillén Galván, Rafael Lemuz López, Juan Escamilla Reyna</i>	
Speech Synthesis Based on Hidden Markov Models and Deep Learning	19
<i>Marvin Coto-Jiménez, John Goddard-Close</i>	
Model-based Algorithm for Belief Revisions between Normal Conjunctive Forms	29
<i>Alma Delia García, Guillermo De Ita, Fernando Zacarias</i>	
Thresholding Approach based on GPU for Facial Expression Recognition.....	41
<i>Jesús García-Ramírez, J. Arturo Olvera-López, Ivan Olmos-Pineda, Georgina Flores-Becerra, Adolfo Aguilar-Rico</i>	
Mathematical Modelling of the Cardiovascular System Haemodynamics	53
<i>Anabel Hernández-Ramírez, Andrés Fraguera-Collar, Rafael Lemuz-López</i>	
Impulsive Noise Removal by Adaptive Mathematical Morphology	65
<i>Marisol Mares Javier, Carlos Guillén Galván, Rafael Lemuz López</i>	

2D Image Segmentation Through the One Dimensional Mumford-Shah Functional

A. Lizbeth Cortés Cortés¹, Carlos Guillén Galván¹, Rafael Lemuz López²,
Juan Escamilla Reyna¹

¹ Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias Físico
Matemáticas, Puebla, México
213570932@alumnos.fcfm.buap.mx, cguillen@fcfm.buap.mx,
jescami@fcfm.buap.mx

² Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la
Computación, Puebla, México
rlemuz@cs.buap.mx

Abstract. Image segmentation is a complex task, since it involves the manipulation of different kind of objects such as functions, domains in \mathbb{R}^2 and curves. In this work we propose a two-dimensional image segmentation method, using the Mumford and Shah's one dimensional technique, performing vertical and horizontal sweeps. This is in contrast to the technique that involves the generalization of the Mumford and Shah functional theory for 2D images.

Keywords: Mumford and Shah functional, Variational Method, Image Segmentation.

1 Introduction

There exist different methods for image segmentation, such as Canny and Sobel, among others. The traditional segmentation methods work by finding the edges of the different objects that are present on the image. The Mumford and Shah method, in addition to this, aims to find a smoothed version of the original image, which makes more attractive its study.

The aforementioned segmentation method, consists on finding a smoothed function and a set of edges (boundary) such that minimize a functional, which is constituted by the combination of the difference between the real image and the smoothed one, the gradient of the smoothed image and the boundary length. There exist developments of the Mumford and Shah methods for image segmentation in one (signals) and two dimensions (images) [5, 6, 3, 2, 7].

Segmenting in one dimension, implies performing an horizontal cut on the image, so that we work only with a signal in \mathbb{R} , which facilitates both the mathematical and computational theories. In [5], the set of edges is determined, i.e., the points where there are intensity changes, and the function between these two points is found. Those are determined by minimizing the functional proposed by the aforementioned authors, analyzing it as a variational problem. First, the set

of edges is fixed, and the smoothed function is varied so, the analytical expression of that function can be found. Then, the set of edges is varied, and in this way, the conditions to determine them are found. With this process, we obtain the segmentation of a tiny fraction of the image, in other words, we obtain a smoothed function through a straight line (horizontal cut), and the discontinuity points (edges) of such function are detected.

The study in [6] is the generalization of the functional proposed in [5], where the one variable functions are changed for two variable functions, with the objective of performing 2D image segmentation. The theory developed for that generalization is on \mathbb{R}^2 , which increases its difficulty with respect to the theory developed on \mathbb{R} . To understand the 2D method, at least it is necessary to have knowledge of measure theory and distributions to understand the special functions of bounded variation and the theory that involves them [3]. As it is mentioned in [2], to avoid to work with those kind of functions, it can be treated only within the space of the L^2 functions, and using a weak norm to measure the approximation between functions. Moreover, it is required certain smoothness on the smoothed image f , therefore, f must belong to a Sobolev space. Another term is the measure of the edges, for this, it is employed the Hausdorff measure of dimension 1.

However, for one dimension it is enough to have knowledge on differential and integral calculus, differential equations and variational calculus. With this contrast, we can ask the question: can the same results be obtained as with the Mumford and Shah method on two dimensions, when the method on one dimension is used?

The objective of this article is targeted at answering this question. We are going to develop the theory for one dimension, to perform image segmentation of 2D images, without going through the mathematical sophistication that requires the generalization of the theory of the functional. This method is applied to every horizontal and vertical cuts of the image, to later, combine both results by averaging the information.

This paper is organized as follows. First, we describe the Mumford and Shah functional, as in [5], followed by its solution via the variational method. Later, the computational algorithm is presented, followed by the experimental results. At the end, the conclusions are stated.

2 The Mumford and Shah Functional

Let μ and ν be fixed parameters. Let $g : [a, b] \rightarrow \mathbb{R}$ be a given continuous function. For each k , the partition $\mathcal{P}[a, b]$ is given by

$$a = a_0 < a_1 < \dots < a_k < a_{k+1} = b.$$

Let $X = \{f : [a, b] \rightarrow \mathbb{R} \mid f \in \mathcal{C}^1([a_i, a_{i+1}])\}$. We define $E : X \times \mathcal{P} \rightarrow \mathbb{R}$ as

$$E(f, \{a_i\}) = \mu^2 \int_a^b (f - g)^2 dx + \sum_{i=0}^k \int_{a_i}^{a_{i+1}} \left(\frac{df}{dx}\right)^2 dx + \mu\nu k. \quad (1)$$

The first term is the approximation from the input signal to the signal that we are looking for, the second term is a measure of the smoothness on (a_i, a_{i+1}) of the signal that we are looking for, and the last term is the cardinality of the set of points $\{a_i\}$ on the open interval (a, b) , which is equal to k .

2.1 Variational Method

We assume that f and $\{a_i\}$ minimize E . The first step is to vary f , to find its value through of the first variation of E and making it equal to zero, [4]. From this, we get the following elliptical problem with boundary values

$$f'' = \mu^2(f - g), \tag{2}$$

$$f'(a_i^\pm) = 0. \tag{3}$$

The solution to this problem can be found with the method of parameter variation, and can be written using the Green's function as follows

$$f(x) = f_\mu(x) + c_{1,i}K_\mu(x - a_i) + c_{2,i}K_\mu(a_{i+1} - x) \tag{4}$$

where

$$f_\mu(x) = \frac{\mu}{2} \int_{a_i}^{a_{i+1}} g(s)e^{-\mu|x-s|} ds \tag{5}$$

and

$$K_\mu(x) = \frac{\mu}{2} e^{-\mu|x|}, \tag{6}$$

$$c_{1,i} = \frac{2}{\mu^2(1 - \alpha_i^2)} (f'_\mu(a_i) - \alpha_i f'_\mu(a_{i+1})), \tag{7}$$

$$c_{2,i} = \frac{2}{\mu^2(1 - \alpha_i^2)} (-f'_\mu(a_{i+1}) + \alpha_i f'_\mu(a_i)), \tag{8}$$

$$\alpha_i = e^{-\mu(a_{i+1} - a_i)}. \tag{9}$$

The function K_μ is known as the Green's function [1]. Moreover, it can be seen that the function f_μ satisfies the differential equation (2).

The next step is to vary one of the points $\{a_i\}$, to obtain their values. After getting the first variation of E and using the fact that f_μ satisfies the equation (2), we get the condition

$$f''_\mu(a_i) \sim 0. \tag{10}$$

Then, the boundary that minimizes the energy E , for a large enough μ , will be the zeros of the second derivative of f_μ .

3 Computational Method

The implementation was done in MATLAB. First, the image was normalized by subtracting the minimum and dividing it by the maximum minus the minimum, in order to achieve consistency in the dynamic range of the pixel intensities. The calculation of the integrals was done with MATLAB's function for the trapezoid rule. The calculation of the derivatives was done with the finite differences method. The algorithm for this is as follows.

Algorithm 1 Program for the Mumford and Shah method for image segmentation in 2D

- 1: $element = row$
 - 2: $a = 1$
 - 3: g equal to $element$ number a ,
 - 4: $n = length(g)$,
 - 5: Partition $x_j = 1, \dots, n$
 - 6: Step size $h = 1$.
 - 7: Calculate points $\{a_i\}$ with $i = 1 \dots k$ such that (10) is satisfied, or equivalently, $f_\mu - g = 0$.
 - 8: Save the values j_i with $i = 1, \dots, k$ such that satisfy $0 \leq j_1 < \dots < j_k < n$ and $a_i \in [x_{j_i}, x_{j_{i+1}}]$.
 - 9: For each interval $[x_{j_{i+1}}, x_{j_{i+1}}]$, set $a_i = x_{j_{i+1}}$ and $a_{i+1} = x_{j_{i+1}}$ and calculate (6), (7), (8), (9), (5), (4) y (1).
 - 10: Find the minimum energy value with the "hill climbing" method: vary j_i while the rest is fixed, and repeat step 9. Save the value j_i when the minimum energy is achieved, and repeat for each j_i .
 - 11: $a = a + 1$
 - 12: Repeat steps 3 to 10 until $a =$ number of rows of the image.
 - 13: $element = column$
 - 14: Repeat steps 3 to 10 until $a =$ number of columns of the image.
 - 15: Calcula the average of steps 12 and 14 for f y j_i .
-

4 Experimental Results

We applied the previously described algorithm to two images, the first is the image known as Barbara, and the second is an image of the famous singer John Lennon. In both cases, we first used the parameters $\mu = 1$ and $\nu = 1$. The results are shown, first the smoothed image after applying the method to each row of the original image, then the smoothed image after applying the method to each column, and lastly the smoothed image consisting on the combination of the previous two. The execution time for each test image is presented. After that, the parameters for the image of John Lennon are modified, and the results are shown.

For the first example, the execution time was 108.31862 min. In Fig. 1 the Barbara input image is shown. The image size is 512×512 . In Fig. 2, the

smoothed images are shown, ordered as described in the previous paragraph. In Fig. 3, the edges that were obtained of each smoothed image are shown.

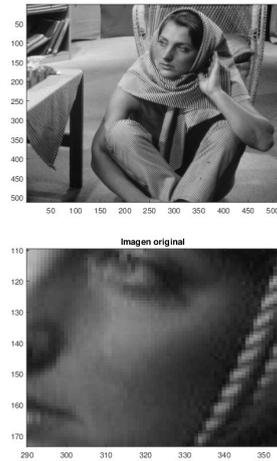


Fig. 1. Barbara original image (512×512)



Fig. 2. From left to right: Smoothed image of Barbara after using the method on each row of the original image, smoothed image after using the method on each column of the original image, resulting smoothed image after combining the previous two

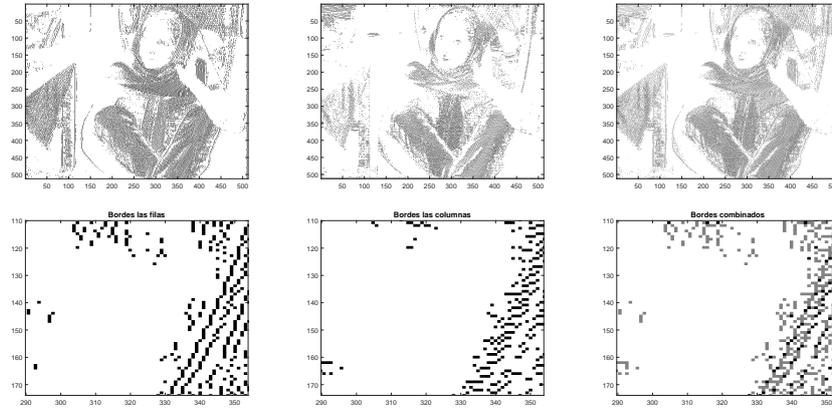


Fig. 3. From left to right: Edges obtained for the Barbara image after using the method on each row of the image, edges obtained after using the method on each column of the image, edges obtained after combination of the two previous results

For the second example, the execution time was 5.49119 min. In Fig. 4, the Lennon input image is shown, which is 204×204 pixel size. In Fig. 5, the smoothed images are shown in the same order as in the previous test. In Fig. 6, the edges obtained from the previous figures are shown.

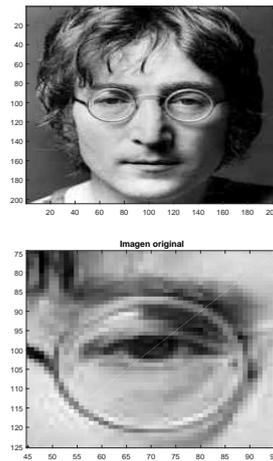


Fig. 4. Original image of John Lennon face (204×204)

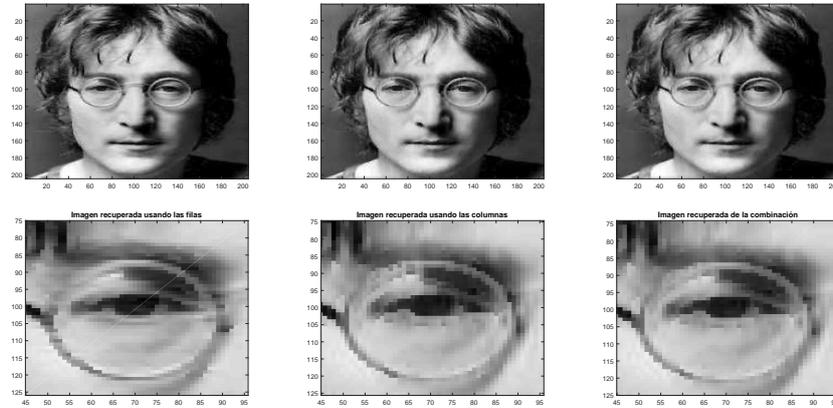


Fig. 5. From left to right: Smoothed John Lennon image after using the method on each row of the original image, smoothed image after using the method on each column of the original image, resulting smoothed image after combining the two previous images

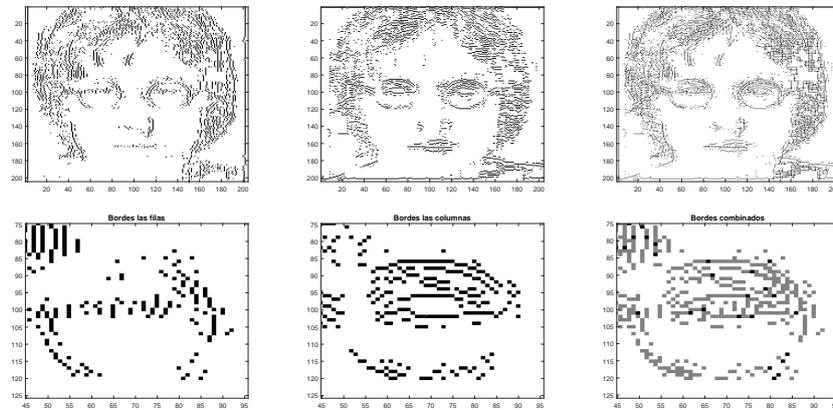


Fig. 6. From left to right: Edges obtained after using the method on each row of the image, edges obtained after using the method on each column of the image, edges obtained from the combination of the previous two

We present the John Lennon image changing the parameters. In Fig. 7 we show the smoothed images resulting from the combination of using the method of rows and columns when $\nu = 1$ and μ takes values of 0.5, 1 and 1.5. In Fig. 8 we show the edges that correspond to the smoothed images with the same values of μ and ν . In the same way, in Fig. 9 we present the smoothed images when

we fix $\mu = 1$, and ν is varied with values of 0.5, 1 and 3. In Fig. 10 we show the corresponding edges.

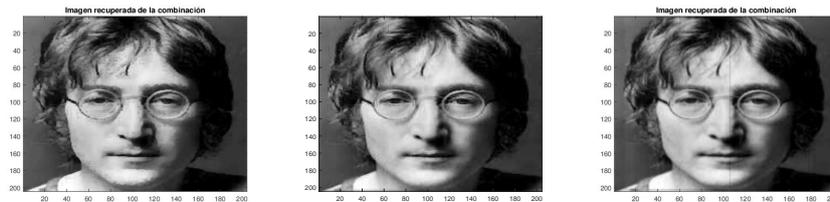


Fig. 7. Smoothed image of John Lennon when: $\mu=0.5$ and $\nu=1$ (left), $\mu=1$ and $\nu=1$ (center), $\mu=1.5$ y $\nu=1$ (right)

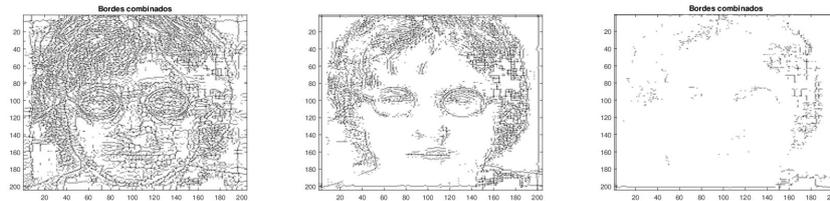


Fig. 8. Edges of the image of John Lennon when: $\mu=0.5$ and $\nu=1$ (left), $\mu=1$ y $\nu=1$ (center), $\mu=1.5$ y $\nu=1$ (right)



Fig. 9. Smoothed image of John Lennon when: $\mu=1$ and $\nu=0.5$ (left), $\mu=1$ and $\nu=1$ (center), $\mu=1$ and $\nu=3$ (right)

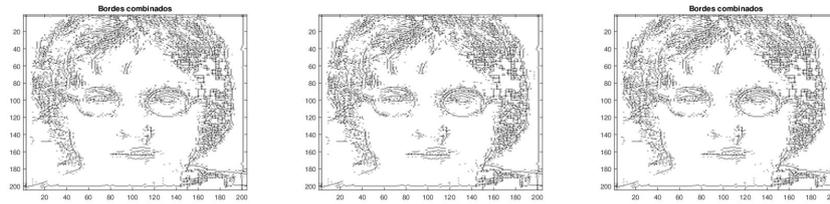


Fig. 10. Edges of the John Lennon image when: $\mu=1$ and $\nu=0.5$ (left), $\mu=1$ y $\nu=1$ (center), $\mu=1$ y $\nu=3$ (right)

It can be seen that the quality of the segmentation varies with respect to the value of the parameters μ and ν . It can be also noted that this variation is more sensitive to the variation of μ than it is to the variation of ν .

5 Conclusions

We achieved the implementation of a computation algorithm for 2D image segmentation based on the Mumford and Shah method for one dimension, i.e., the procedure for one dimension was generalized to two dimensions. The advantage of this technique, is that the theory is kept on \mathbb{R} , which is much less complex than working directly on \mathbb{R}^2 .

The obtained results by applying this segmentation method, are the smoothed image that approximates to the input image, and the edges of that image. This methods has a high computational cost, since the execution time is large, which depends on the size of the image and number of edges. Despite of this, a good representation is obtained, although it depends on the parameters μ and ν .

The method still needs to be improved for it to be compared to the method designed for 2D, however, is a good start. In [7], the segmentation done for the Barbara image, with the approximation of Ambrosio and Tortorellini, can be found.

In a future study, we will make an optimization to the algorithm to decrease the execution time, in addition to combine this with other tools to improve the segmentation. We will also analyze to measure the performance of the method with noisy images, with the Peak signal-to-noise ratio (PSNR) metric[8] and Structural SIMilarity (SSIM) index[9].

Acknowledgments

This research was facilitated through a doctoral scholarship from CONACyT under Grant 386423 and the PRODEP network of mathematical and computational modeling in medicine and demographic dynamics.

We thank the reviewers for their contributions to improve this research and writing of the manuscript.

References

1. Boyce, W.E., DiPrima, R.C.: Ecuaciones diferenciales y problemas con valores en la frontera. Limusa (2000)
2. David, G.: Singular Sets of Minimizers for the Mumford-Shah Functional. Birkhauser Verlag (2005)
3. G., A., P., K.: Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations. Springer-Verlag (2006)
4. Gelfand, I.M., Fomin, S.V.: Calculus of Variation. Prentice-Hall Inc. (1963)
5. Mumford, D., Shah, J.: Boundary detection by minimizing functionals. In: IEEE Proceedings on Computer Vision and Pattern Recognition. pp. 22–26. San Francisco, California (1985)
6. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics* 42, 577–685 (1989)
7. Scherzer, O.: Handbook of mathematical methods in imaging. Springer (2011)
8. Wang, Z., Bovik, A.C.: Mean squared error: love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine* 26, 98–117 (2009)
9. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 600–612 (2004)

Speech Synthesis Based on Hidden Markov Models and Deep Learning

Marvin Coto-Jiménez^{1,2}, John Goddard-Close²

¹ University of Costa Rica, San José, Costa Rica
`marvin.coto@ucr.ac.cr`

² Autonomous Metropolitan University, México D.F., México
`jgc@xanum.uam.mx`

Abstract. Speech synthesis based on Hidden Markov Models (HMM) and other statistical parametric techniques have been a hot topic for some time. Using this techniques, speech synthesizers are able to produce intelligible and flexible voices. Despite progress, the quality of the voices produced using statistical parametric synthesis has not yet reached the level of the current predominant unit-selection approaches, that select and concatenate recordings of real speech. Researchers now strive to create models that more accurately mimic human voices. In this paper, we present our proposal to incorporate recent deep learning algorithms, specially the use of Long Short-term Memory (LSTM) to improve the quality of HMM-based speech synthesis. Thus far, the results indicate that HMM-voices can be improved using this approach in its spectral characteristics, but additional research should be conducted to improve other parameters of the voice signal, such as energy and fundamental frequency, to obtain more natural sounding voices.

Keywords: LSTM, HMM, Speech Synthesis, Statistical Parametric Speech Synthesis, Deep Learning.

1 Introduction

Speech synthesis is the process of creating artificial intelligible speech from a text, to propitiate human-machine interaction. Through this process, text-based information originated from computers, cell phones, and other technological devices can be transferred into speech. Since speech is the most common and natural way of communication for most human beings, this process is invaluable.

Due to its variation in speaking styles and potential for customization, speech synthesis based on HMM (also called statistical parametric speech synthesis) has been established as a viable alternative to dominant models, that are based on the concatenation of audio units [1].

In dominant models, audio waves of recording speech are handled directly, concatenating the wave segments to form new sentences. On the contrary, statistical parametric synthesis is based on the use of mathematical models that can learn and then reproduce parameters representing speech, for example HMM.

Recently, a small group of researchers [2–6], motivated by the success of deep learning algorithms in speech recognition, have reported encouraging results in adapting such algorithms in speech synthesis. As a result, a new possibility of developing artificial voices emerged, such that the original parameters could be kept while using new mathematical models that more accurately reproduce the characteristics of a human voice.

The reports made with these new techniques are preliminary due to restricted access to audio samples and to programs that reproduce these experiments and verify the results. This proposal aims to analyze such deep learning algorithms at different levels in statistical parametric speech synthesis, and develop techniques for using them in the process of creating higher quality artificial voices to extend their use and create new applications.

The rest of this paper is organized as follows: Section II describes the Research Problem; Section III presents the Research Methodology; Section IV discusses the expected Main Contribution; Section V reports the Results achieved so far; and finally Section VI presents the conclusions.

2 Background

Hidden Markov Models can be described from a Markov process, in which state transitions are given by a stochastic process. A second stochastic process models the emission of symbols when it comes to each state. In Figure 1, a representation of a left to right HMM is shown, where there is a first state to the left from which transitions can occur to the same state or to the next on the right, but not in reverse direction. In this p_{ij} represents the probability of transition from state i to state j , and O_k represents the observation emitted in state k . In speech synthesis and recognition, O_k are vectors of parameters representing speech, with fundamental frequency, spectral and duration information.

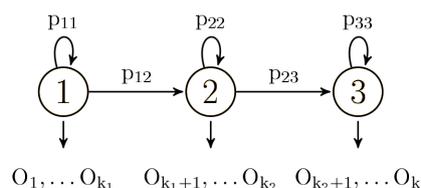


Fig. 1. Left to right example of an HMM with three states

HTS [7] is the only consolidated tool for using HMM in speech synthesis. The HTS automates the processes of extracting parameters (fundamental frequency, spectral and duration), training the HMM, and creating decision trees for search. Thus, it can search and find the most appropriate HMM to generate the parameters of the new phrases and synthesize a speech waveform.

In contrast to traditional techniques for artificial voice production, statistical parametric speech synthesis is based on mathematical models that learn to generate parameters which can build sound waves of speech, allowing for the pronunciation of any arbitrary text. This requires the extraction of an adequate representation of parameters that can model a person's voice and the combination and generation of new phrases that preserve the features of the original voice.

To complete this task, the process requires input into a database of natural voice recordings and the text of the new phrases that requires pronunciation using the artificial voice. Statistical parametric speech synthesis has many advantages over concatenative approaches, such as:

- Greater flexibility, because it is possible to modify characteristics of the synthesized speech, combining voices and create new ones manipulating the parameters.
- Small footprint, because only the parametric representations have to be stored and processed.
- Small requirements of database size.

Despite these advantages, its quality is not yet comparable to the best systems based on concatenation of speech units, so its use in applications is still not as widespread. The main reasons are [1]:

- Model of speech production: The reconstruction of speech waveform is based on a source-filter model, where the sounds with fundamental frequency are generated from a pulse train, and sounds without fundamental frequency are generated from white noise. This is considered a simplistic model in comparison with the production of human speech.
- Over smoothing: HMMs are trained with examples of speech that are segmented into phonemes and then averaged, which results in some loss of information. This results in smaller variability in contours of fundamental frequency and spectra.
- Mathematical modeling: It has been questioned whether the HMM may suitably represent natural speech. Some extensions of the technique as Global variance [8], Trajectory HMM [9] and autoregressive HMM [10] have presented improvements, but still need to prove its usefulness in the context of more natural and expressive speech.

3 Research Problem

The adoption of HMMs with Gaussian distributions to model parameters has emerged from its proven efficiency in the topic of speech recognition, previously developed since the 1980s in works like [11, 12]. Subsequently, there have been some improvements in the area of recognition. For example, the Adaptation Technique, where average models of Gaussian distributions are linearly transformed to improve recognition in new speakers [13, 14], produced benefits in its

equivalence in speech synthesis, such as the creation of new voices. Recently, researchers have explored the use of new models and strategies for speech recognition, with encouraging results. Specifically, deep learning algorithms, based on deep neural network architectures stacking many hidden layers, have been highlighted. These implementations can be classified into two types:

1. Type I: Replacing the HMM with deep neural networks, so that all the acoustic representation of speech is done with the new models. Experiences using models such as: Deep Artificial Neural Networks (DNN) [15, 16] and recurrent artificial neural networks (RNN) [17], including the special case of LSTM [18] networks.

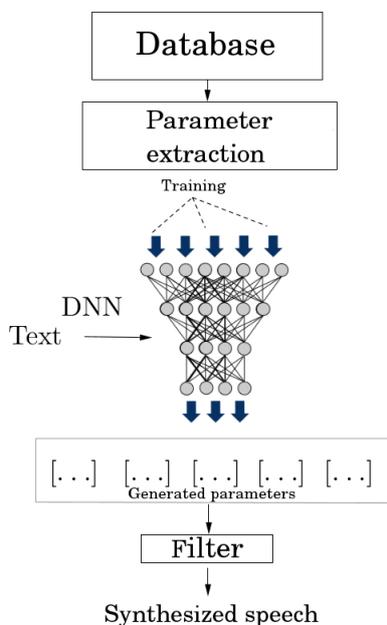


Fig. 2. Replacing the HMM by deep learning networks for parametric speech synthesis (Type I)

2. Type II: Adaptation or modification of the HMM models to improve recognition: Similar to the technique of adaptation, deep learning algorithms to improve HMM in recognition has been explored, and thereby increase the recognition rate. This is an addition to the models that have been tested for decades, without completely replacing the HMM. For example, in [19–21], where some of the elements of the voice are transformed.

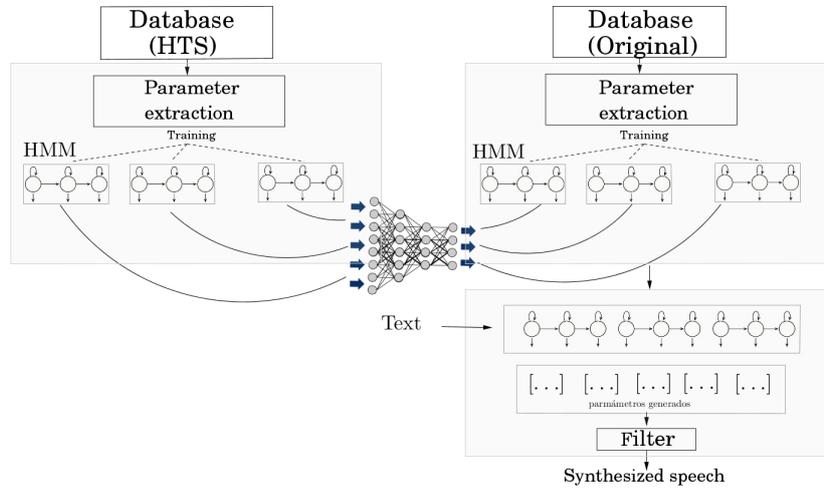


Fig. 3. Transforming HMM with deep learning algorithms for parametric speech synthesis (Type II)

From these results and following the path of incorporating mathematical models in speech synthesis that have been successful previously in recognition, emerges the possibility of improving the quality of artificial voices with the addition of deep learning algorithms. The two types of strategies are shown in Figures 2 and 3, in which HMM are substituted with deep learning networks (Type I) or are incorporated into transformation models (Type II).

Improvements that these algorithms can bring to speech synthesis are in at least one of the quality characteristics of speech synthesis based on HMM, for example:

- Greater naturalness and intelligibility
- Greater preference by users
- Greater capacity to produce emotive voices
- Lesser oversmoothing in spectral and f_0
- Smaller requirements on database size
- Less complexity in the processes of creating voices

4 Research Methodology

Multiple steps are required to produce synthesized speech from an arbitrary text, using deep learning algorithms. From language modeling to evaluating results, arises the need to address the following stages:

4.1 Creation of Voice Databases

The first experiments carried out to synthesize voice, taking into account HMM based speech synthesis and deep learning, have been conducted on large databases with more than five hours of recordings. Therefore, it is necessary to analyze possible sources of data and appropriate phrases for use them in synthesis.

After the selection and processing of the voice recordings, it is necessary to separate sentences and carefully transcribe each one of them. The next step is to codify the features of the text that are necessary and sufficient enough to predict prosodic aspects and produce more natural speech across various contexts.

The most common features extracted from the speech signal arises from those used in speech recognition: Mel Frequency Cepstral Coefficients (MFCC) and f_0 . A vector of parameters, and their first and second derivative discrete-approximation are extracted from each frame of 10 ms.

4.2 Segmentation of Audio and Parameters Extraction

When processing the phonetic units of speech that can be used in statistical parametric speech synthesis with deep learning algorithms, it is necessary to have methods to establish their boundaries within the recorded phrases, and then extract them into the corresponding parametric representation. There are several possibilities to use parameters that must be analyzed according to the viability of using them on the algorithms to implement.

In both Type I and Type II systems, coding can make a noticeable difference. It should be noted that there are also restrictions on the coding complexity raised from processing time and computer memory.

4.3 Algorithm Implementation

After having the appropriate representation of the speech signal in parameters, extensive experimentation should be performed on available systems to implement some of the proven deep learning algorithms in speech recognition: DNN, LSTM, BLSTM and RBM, and determine its feasibility to be incorporated into the development speech synthesis, in both Type I and Type II systems.

4.4 Evaluation of Results

There are different approaches to solving the problem of evaluating objectively synthesized speech. Since software is unavailable for this purpose, theory should be studied thoroughly and then implemented based on the proposed measures.

The subjective assessment, based on listeners who evaluated according to certain scales of perception, are often used to complement and validate the final audio results.

5 Main Contribution

This will be the first time that implementation of deep learning algorithms for both Type I and Type II speech synthesis systems will be conducted and documented in detail. With the results, we can provide a comparative analysis between both systems and propose full text to speech conversion systems with those models that have better functionality.

The required experimentation includes modeling phonetic units that have not previously been considered in parametric speech synthesis for the case of Spanish, such as syllables or groups of phonemes, which may constitute a contribution even for the synthesis based on HMM.

It is expected to obtain the first artificial voices in Spanish language using such algorithms, including the possibility of incorporating unused algorithms in other languages, such as Convolutional Networks, Deep Auto Encoders and LSTM.

We believe that the application of these algorithms can improve at least one aspect of quality of the voices that are currently obtained with statistical parametric speech synthesis based on HMM. Thereby enabling the widespread use of this technology in innovative applications, such as changing a voice accent and speech synthesis with emotions, for use in applications such as speech to speech translation in real time between different languages.

6 Results Achieved

Our first results with statistical parametric speech synthesis using HTS to produce voices in Mexican Spanish would have shown the advantages of using this technique, and have been documented in [22–24]. Those results were reported without any deep learning algorithms involved, and showed the improvement opportunities in the quality of the voices. Later, we experimented with Type II systems using Long Short-term Memory (LSTM) deep neural networks. The input of the network is the artificial voice produced with HTS, and the output is the corresponding natural voice, thus the network is trained to bring the sound of the synthesized speech closer to the natural voice. Figure 4 shows the current system development.

This is equivalent to find a mapping $y = f(x)$ between a degraded set of n -dimensional parameters x , from HTS voice, to a clean n -dimensional parameters y of the natural voice. This is a similar approach to denoising in speech enhancement and speech recognition, with recent implementations of LSTM with successful results [25].

With this scheme, there have been notable improvements in the parameters corresponding to the spectrum of the signals. For example, Figure 5 shows the trajectory of one of the spectral parameters used, and shows how the LSTM improved artificial speech more closely resembles natural speech than the same parameter in the HTS voice.

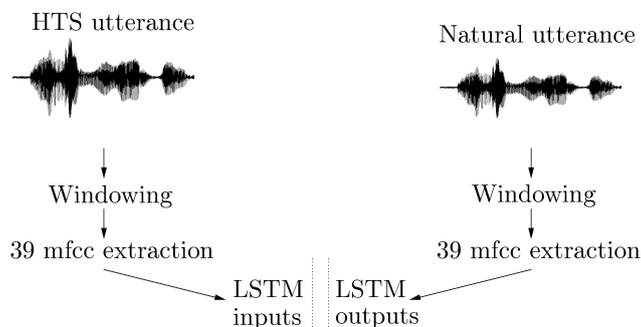


Fig. 4. Proposed system. HTS and Natural utterances are aligned frame by frame

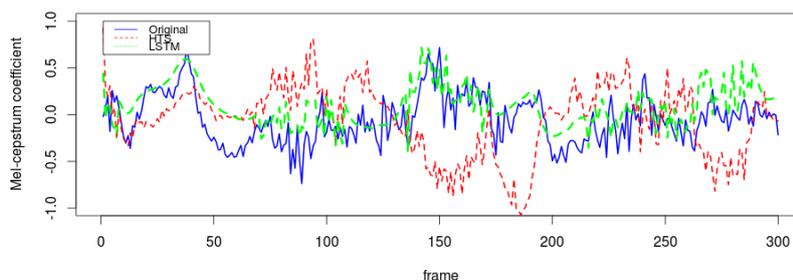


Fig. 5. Illustration of enhancing the 5th mel-cepstral coefficient trajectory by LSTM enhancement

Observations on other objective measures such as Mel-Cepstral Distance and similarity of spectrograms, also show significant improvements, indicating the improvement of the voice processed with LSTM.

7 Conclusions

We have presented a proposal for the implementation of deep learning algorithms to improve the results achieved so far in statistical parametric speech synthesis based on Hidden Markov Models.

The strategy is based on preparing frame aligned data of synthesized and natural speech. Further, we propose to use the algorithms to model the transformation of parameters of the artificial voice generated from the HTS system, the corresponding voices of the natural speech.

The results show improvement of spectral features of the voice in objective evaluation, but have to be tested with subjective assessments. New strategies for data modeling, algorithms and configurations have to be explored in order to improve other aspects of the speech signal, such as the fundamental frequency, energy and aperiodic coefficients that can further improve the sound of the artificial voice to a natural voice.

Acknowledgments

This work was supported by the SEP and CONACyT under the Program SEP-CONACyT, CB-2012-01, No.182432, in Mexico, as well as the University of Costa Rica in Costa Rica.

References

1. Tokuda, K., Nankaku, Y., Toda, T., Zen, H., Yamagishi, J., Oura, K.: Speech synthesis based on hidden markov models. In: Proceedings of the IEEE, vol. 101(5), pp. 1234–1252 (2013)
2. Zen, H., Senior, A., Schuster, M.: Statistical parametric speech synthesis using deep neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7962–7966 (2013)
3. Zhen-Hua, L., Deng, L., Dong, Y.: Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis. IEEE Transactions on Audio, Speech, and Language Processing 21(10), 2129–2139 (2013)
4. Shiyin, K., Qian, X., Hsiang-Yun, M.: Multi-distribution deep belief network for speech synthesis. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8012–8016 (2013)
5. Fan, Y., Qian, Y., Xie, F.L., Soong, F.K.: TTS synthesis with bidirectional LSTM based recurrent neural networks. Interspeech 1964–1968 (2014)
6. Zhizheng, W., Valentini-Botinhao, C., Watts, O., King, S.: Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4460–4464 (2015)
7. HMM-based Speech Synthesis System (HTS). <http://hts.sp.nitech.ac.jp/>
8. Tomoki, T., Tokuda, K.: A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. IEICE TRANSACTIONS on Information and Systems 90(5), 816–824 (2007)
9. Zen, H., Tokuda, K., Kitamura, T.: Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. Computer Speech & Language, 21(1), 153–173 (2007)
10. Shannon, M., Zen, H., Byrne, W.: Autoregressive models for statistical parametric speech synthesis. IEEE Transactions on Audio, Speech, and Language Processing 21(3), 587–597 (2013)
11. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. In: Proceedings of the IEEE, vol. 77(2), pp. 257–286 (1989)

12. Wellekens, C.: Explicit time correlation in hidden Markov models for speech recognition. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 12, pp. 384–386 (1987)
13. Giuliani, D., Omologo, M., Svaizer, P.: Experiments of speech recognition in a noisy and reverberant environment using a microphone array and HMM adaptation. In: Proceedings of the Fourth International Conference on Spoken Language, ICSLP 96, vol. 3, pp. 1329–1332 (1996)
14. Flores, J.A., Young, S.J.: Continuous speech recognition in noise using spectral subtraction and HMM adaptation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 409–412 (1994)
15. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Kingsbury, B.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine* 29(6), 82–97 (2012)
16. Dahl, G.E., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing* 20(1), 30–42 (2012)
17. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649 (2013)
18. Graves, A., Eck, D., Beringer, N., Schmidhuber, J.: Biologically plausible speech recognition with LSTM neural nets. In: *Biologically Inspired Approaches to Advanced Information Technology*, Springer Berlin Heidelberg, pp. 127–136 (2004)
19. Seide, F., Gang, L., Dong, Y.: Conversational Speech Transcription Using Context-Dependent Deep Neural Networks. *Interspeech*, 437–440 (2011)
20. Dong, Y., Chen, X., Deng, L.: Factorized deep neural networks for adaptive speech recognition. In: *International workshop on statistical machine learning for speech processing* (2012)
21. Dong, Y., Yao, K., Su, H., Li, G., Seide, F.: KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7893–7897 (2013)
22. Coto-Jiménez, M.: Síntesis estadística paramétrica de voz. Master thesis. Universidad Autónoma Metropolitana, Unidad Iztapalapa. México D.F., México (2014)
23. Coto-Jiménez, M., Goddard-Close, J., Martínez-Licona, F.M.: Quality Assessment of HMM-Based Speech Synthesis Using Acoustical Vowel Analysis. In: *Speech and Computer*. Springer International Publishing, pp. 368–375 (2014)
24. Coto-Jiménez, M., Martínez-Licona, F.M., Goddard-Close, J.: Acoustic Vowel Analysis in a Mexican Spanish HMM-based Speech Synthesis. *Research in Computing Science* 86, 53–62 (2014)
25. Coto-Jiménez, M., Goddard-Close, J., Martínez-Licona, F.M.: Improving Automatic Speech Recognition Containing Additive Noise Using Deep Denoising Autoencoders of LSTM Networks. In: *18th International Conference on Speech and Computer SPECOM* (Forthcoming) (2016)

Model-based Algorithm for Belief Revisions between Normal Conjunctive Forms

Alma Delia García, Guillermo De Ita, Fernando Zacarias

Benemerita Universidad Autonoma de Puebla, Computer Sc. Dpt., Puebla, México
deliakzy@gmail.com, deita@cs.buap.mx, fzflores@yahoo.com.mx

Abstract. We consider a knowledge base (KB) K and a new information ϕ , both expressed in conjunctive form (CF), and present here, a novel, deterministic and correct algorithm for belief revision of ϕ in K . We denote our revision operator as: $K' = K \circ \phi$. We introduce a novel logical binary operator Ind between two conjunctive forms, such that $Ind(\phi, K)$ generates also a conjunctive form. The operator $Ind(\phi, K)$ works building independent clauses with the clauses of K , and whose falsifying assignments of the resulting formula cover exactly the space of assignments $Fals(\phi) - Fals(K)$, this is essential for performing the process of belief revision $K' = K \circ \phi$, and where $K' \models \Phi$. Furthermore, our proposal satisfies the KM postulates. We also present the correctness proof of our belief revision method, and the analysis of its time complexity.

Keywords: Propositional Inference, Belief Revision, Model Based Inference, Postulates KM.

1 Introduction

A widely accepted reference framework for reasoning in intelligent systems is the systems approach based on knowledge bases. In this case, the sentences are stored in a Knowledge Base (KB) provided with a reasoning mechanism [5]. A fundamental challenge of these systems is the automation of deductive reasoning from the KB. Propositional deductive reasoning is generally summarized as follows: given a KB K , which contains knowledge about a domain ("the world"), and ϕ a statement representing the query that captures the current situation, both expressed in propositional logic, the goal is to decide whether K implies ϕ (in symbols: $K \models \phi$), which is known as the *problem of propositional entailment*.

The propositional implication is an important task in problems such as estimating the degree of belief revision and updating of beliefs, working with abductive explanations, and many other procedures in applications of Artificial Intelligence. For example, when working in planning, designing multi-agent systems, logical diagnosis, approximate reasoning, among other applications [4], [7]. In general, the problem of logical implication is a difficult challenge in the area of automatic reasoning, and turns out to be a problem in the class Co-NP complete, even in the propositional case [9].

Belief revision problem consists in incorporate new beliefs to a knowledge base (KB) already established, changing as little as possible the original beliefs and maintaining consistency of the KB. This article shows that the use of falsifying patterns clauses help to determine whether an conjunctive form (CF) is inferred from another CF, and therefore, it allows us to construct an algorithm for belief revision between CF's.

In short, the main contributions of our work are:

- We propose a method that works on the set of falsifying assignments of the involved formulas, in order to review: $K \models \Phi$.
- We introduce a logical operator between two terms, $Ind(\varphi_i, C_j)$, resulting in a CF Fs , such that $Fals(Fs) = Fals(\varphi_i) - Fals(C_j)$.
- We show that our proposal for the belief revision is correct, and holds the Katsuno and Mendelzon postulates.
- $Ind\ operator(\varphi_i, C_j)$ was implemented to work in linear time on number of variables involved, and it is the base of our procedure for belief revision.
- Although the operator $Ind(\varphi_i, C_j)$ performs efficiently, the number of clauses in Fs , that allows $(K \wedge Fs) \models \phi$, can lead to an exponential growth (the order $O(2^{(n-\min\{|\varphi_i|:\varphi_i \in \phi\})})$).

1.1 State of the Art

The paradigm best known for belief revision is the AGM paradigm [1]. Subsequently, Mendelzon and Katsuno [8] unified the different belief revision approaches to semantic, and reformulated the AGM postulates, which were called now, KM postulates. Subsequently, Darwiche and Pearl [3] proposed the iterated revision, where his proposal establishes a representation based on model assumptions.

There are some proposals for belief revisions based on models, and they are identified by the name of their authors; Dalal, Satoh, Winslett, Borguida and Forbus [11]. Dalal [2] operator suggests to review, based on the minimum Hamming distance between satisfied interpretations, and extend the distance between interpretations and bases. In practice, this proposal involves the calculation of the set of models, which is very expensive. One of the drawbacks of the approach from Dalal is limited to the case of consistent knowledge bases. Therefore in [13] is proposed a new method for computing Dalal's revision that avoids the computation of belief bases models, but it works only on disjunctive normal forms.

The Satoh's proposal [14] is similar to Dalal, with the difference that the distance between two models is defined as the set of different literals between both models. In the case of Winslett, the proposal is based on a comparison between all possible consistent maximum systems. The Borguida and Forbus' proposals are similar of the Winslett, with the difference that Borguida considers incompatible models, and Forbus use the Hamming distance. More recently, the belief revision has gained attention in the framework of symbolic logic, including Horn fragments, and many operators for belief revision have been proposed according to syntactic or semantic points of view [10, 12, 14].

2 Preliminaries

Let $X = \{x_1, \dots, x_n\}$ be a set of n Boolean variables. A *literal* denoted as *lit* is, a variable x_i or a denied variable $\neg x_i$. As usual, each $x \in X$, $x^0 = \neg x$ y $x^1 = x$. A *clause* is a disjunction of different literals. For $k \in N$, a *k-clauses* is a clause with exactly k literals, and *($\leq k$)-clause* is a clause with at most k literals. A *phrase* is a conjunction of literals. A *k-phrase* is a phrase with exactly k literals. A variable $x \in X$ appears in a clause C (or phrase) if x or $\neg x$ is an element of C .

A *conjunctive normal form (CF)* is a conjunction of clauses, and *k-CF* is a *CF* containing only *k-clauses*. A *disjunctive normal form (DF)* is a disjunction of sentences, and *k-DF* is a *DF* containing only *k-phrases*. A *CF* F with n variables is a *n-ary* Boolean function $F: \{0, 1\}^n \rightarrow \{0, 1\}$. Rather, any Boolean function F has infinitely many equivalent representations, among these, some in *CF* and *DF*.

We denote with Y any of the basic logic elements that we are using, such as a literal, a clause, a phrase, a *DF* or a *CF*. $v(Y)$ denotes the set of variables involved in the object Y . For example, $v(\neg x_1 \vee x_2) = \{x_1, x_2\}$. While $lit(Y)$ denotes the set of literals involved in object Y . For example, if $X = v(Y)$ then $lit(Y) = X \cup \neg X = \{x_1, \neg x_1, \dots, x_n, \neg x_n\}$. We also use $\neg Y$ as the negation operator on the object Y . We denote to $\{1, 2, 3, \dots, n\}$ by $[[1, n]]$, and the cardinality of a set A by $|A|$.

An *assignment* s for a formula F is a Boolean mapping $s : v(F) \rightarrow \{1, 0\}$. An assignment s can also be considered as a non-complementary set of literals: $l \in s$ if and only if s assigns true to l and $\neg l$ false. s is a partial *assignment* for the formula F when s has determined a logical value only to variables of a proper subset of F , namely $s : Y \rightarrow \{1, 0\}$ and $Y \subset v(F)$. A *CF* F is satisfied by an assignment s if each clause F is satisfied by s ; A model of F is an assignment on $v(F)$ satisfying F .

A phrase f is satisfied by an assignment s , if $f \subseteq s$. Otherwise, s falsifies f . A *DF* F is satisfied by s if a sentence in F is satisfied by s . F is contradicted by s if all sentences in F are contradicted by s . Denote by $SAT(F)$ to the set of assignment in $S(F)$ which they are models for F . $Fals(F)$ denotes the set of assignments in $S(F)$ that falsifies F .

3 Inference Between Conjunctive Forms

In this section we analyze the computational complexity of the entail problem between conjunctive forms: $CF \models CF$. As K and ϕ are in *CF*, the falsifying assignments $Fals(K)$ y $Fals(\phi)$ can be calculated efficiently [5]. Using falsifying strings as the base for reviewing $K \models \phi$, that is equivalent to check whether $SAT(K) \subseteq SAT(\phi)$, or that: $Fals(\phi) \subseteq Fals(K)$. The result of applying the operator of belief revision between a KB K and the new evidence ϕ is denoted as $K' = K \circ \phi$.

When $K \models \phi$ then $K' = K \circ \phi = K$. If $K \not\models \phi$ then $Fals(\phi) \not\subseteq Fals(K)$, which implies that there is a set of assignments S such that $S \subseteq Fals(\phi)$ y $S \not\subseteq Fals(K)$.

$Fals(K)$. If $K \not\models \phi$, then $S = (Fals(\phi) - Fals(K)) \neq \emptyset$. In this case, our belief revision method works building such set S , which allows to build a new CF Fs , such that $S = Fals(Fs)$ and $K' = (K \wedge Fs)$, and it holds $K \models \phi$.

The proposed method obtains $S = (Fals(\phi) - Fals(K))$ as a set of falsifying strings, leading us to build a CF Fs directly, where $S = Fals(Fs)$ and such that $K' = K \wedge Fs$ is a new CF with less information than K (because K' has more clauses than K), in fact, it holds that if $S \neq \emptyset$ then $Fals(K) \subset Fals(K')$, and therefore, $SAT(K') \subset SAT(K)$.

3.1 Construction of Independent Sets of Clauses

Let K be a CF, i.e., $K = \bigwedge_{i=1}^m C_i$, where each $C_i, i = 1, \dots, m$ is a disjunction of literals. The set of assignments forming $Fals(C_i)$ can be represented through a string A_i that consists of $\{0, 1, *\}$. If $C_i = \{x_{i_1} \vee \dots \vee x_{i_k}\} \in K$, then the value; for each position from $i_1 - th$ to $i_k - th$ of the string A_i , has to falsify the literals of C_i . If $x_{i_j} \in C_i$ then the $i_j - th$ element of A_i is set to 0. If $\neg x_{i_j} \in C_i$ then the $i_j - th$ element of A_i is set to 1. The variables in $v(K)$ which do not appear in C_i are represented by the symbol *, meaning that they could take any logical value $\{0, 1\}$. In this way, the string A_i of length $n = |v(K)|$ represents the set of assignments falsifying the clause C_i . We will denote the formation of the string representing $Fals(C_i)$ as $A_i = string(Fals(C_i))$. It is easy to build $Fals(K)$ since each clause C_i determines a subset of falsifying assignments of K . The following lemma expresses how to form the falsifying set of assignments of a CF.

Lemma 1. *Given a CF $K = \bigwedge_{i=1}^m C_i$, which holds $Fals(K) = \bigcup_{i=1}^m \{\sigma \in S(K) \mid Fals(C_i) \subseteq \sigma\}$*

Definition 1. [6] *Given two C_i and C_j clauses, if they have at least one complementary literal, will be called independent clauses. Otherwise, it is said that both are dependent clauses.*

Two independent clauses C_i and C_j have complementary pair of literals, therefore their falsifying assignments also must have complementary literals, that is, $Fals(C_i) \cap Fals(C_j) = \emptyset$.

Definition 2. *Let $K = \{C_1, C_2, \dots, C_m\}$ be a CF. K is called independent if for any pair of clauses $C_i, C_j, \in K, i \neq j$, the property of independence is met.*

Definition 3. *Given two falsifying strings A and B each of length n , if there is an $i \in \{1, \dots, n\}$ such that $A[i] = x$ and $B[i] = 1 - x, x \in \{0, 1\}$, it is said that they have the independence property. Otherwise, we say that both strings are dependent.*

Notice that falsifying strings for independent clauses have complementary values (0 and 1) in at least one of their fixed values.

Let $F = \{C_1, C_2, \dots, C_m\}$ be a CF, $n = |v(F)|$. Let $C_i, i \in [[i..m]]$ be a clause in F and $x \in v(F) \setminus v(C_i)$ be any variable, we have that

$$C_i \equiv (C_i \vee \bar{x}) \wedge (C_i \vee x) \quad (1)$$

Furthermore, this reduction preserves the number of falsifying assignments of C_i with respect to F , since $\#FAL(C_i) = 2^{n-|C_i|} = 2^{n-(|C_i|+1)} + 2^{n-(|C_i|+1)} = \#FAL((C_i \vee \bar{x}) \wedge (C_i \vee x))$, because $(C_i \vee \bar{x})$ and $(C_i \vee x)$ are two independent clauses.

Definition 4. Given a pair of dependent clauses C_1 and C_2 , if $lit(C_1) \subseteq lit(C_2)$ we say that C_2 is subsumed by C_1 .

If C_1 subsumes C_2 then $FAL(C_2) \subseteq FAL(C_1)$. On the other hand, if C_2 is not subsumed by C_1 and they are dependents, there is a set of indices $I = \{1, \dots, p\} \subseteq \{1, \dots, n\}$ such that for each $i \in I, x_i \in C_1$ but $x_i \notin C_2$. There exists a reduction to transform C_2 to be independent with C_1 , we call this transformation as the *independent reduction* between two clauses that works as follows: let C_1 and C_2 be two dependent clauses. Let $\{x_1, x_2, \dots, x_p\} = Lit(C_1) \setminus Lit(C_2)$. By (1) we can write: $C_1 \wedge C_2 = C_1 \wedge (C_2 \vee \neg x_1) \wedge (C_2 \vee x_1)$. Now C_1 and $(C_2 \vee \neg x_1)$ are independent. Applying (1) to $(C_2 \vee x_1)$:

$$C_1 \wedge C_2 \equiv C_1 \wedge (C_2 \vee \neg x_1) \wedge (C_2 \vee x_1 \vee \neg x_2) \wedge (C_2 \vee x_1 \vee x_2)$$

The first three clauses are independent. Repeating the process of making the last clause independent with the previous ones, until x_p is considered; we have that $C_1 \wedge C_2$ can be written as:

$$C_1 \wedge (C_2 \vee \neg x_1) \wedge (C_2 \vee x_1 \vee \neg x_2) \wedge \dots \wedge (C_2 \vee x_1 \vee x_2 \vee \dots \vee \neg x_p) \wedge (C_2 \vee x_1 \vee x_2 \vee \dots \vee x_p).$$

The last clause contains all literals of C_1 , so it is subsumed by C_1 , and then

$$C_1 \wedge C_2 \equiv C_1 \wedge (C_2 \vee \neg x_1) \wedge (C_2 \vee x_1 \vee \neg x_2) \wedge \dots \wedge (C_2 \vee x_1 \vee x_2 \vee \dots \vee \neg x_p) \quad (2)$$

We obtain on the right side of (2) an independent set of $p + 1$ clauses which we denote as *indep_reduction*(C_1, C_2).

We will use the independent reduction between two clauses C_1 and φ (or between their respective falsifying strings) to define:

$$Ind(C, \varphi) = \begin{cases} \varphi & \text{If } \varphi \text{ and } C \text{ are independent} \\ \emptyset & \text{If } Lit(C) \setminus Lit(\varphi) = \emptyset \\ indep_reduction(C, \varphi) - C & \text{in other case} \end{cases} \quad (3)$$

It is straightforward to redefine the operator *Ind* in terms of the falsifying strings representing $FAL(C)$ and $FAL(\varphi)$. The operation $Ind(C, \varphi)$ forms a conjunction of clauses whose falsifying assignments are exactly $FAL(\varphi) - FAL(C)$.

Theorem 1. If φ and C are two clauses, then $FAL(Ind(C, \varphi)) = FAL(\varphi) - FAL(C)$.

Proof. If $Ind(C, \varphi) = \emptyset$ then $FAL(\varphi) \subseteq FAL(C)$, so $FAL(\varphi) \setminus FAL(C) = \emptyset$. Now, we assume that $Ind(C, \varphi) \neq \emptyset$. Let s be an assignment such that $s \in FAL(Ind(C, \varphi))$. We will show that $s \in FAL(\varphi)$ and $s \notin FAL(C)$. If $s \in FAL(Ind(C, \varphi))$ then s falsifies φ because each clause in $Ind(C, \varphi)$ has the form $(\varphi \vee R)$, where R is a disjunctive set of literals (possibly R is empty). If s falsifies $(\varphi \vee R)$ then s has to falsify φ and thus $s \in FAL(\varphi)$. On the other hand, each clause $(\varphi \vee R) \in Ind(C, \varphi)$ is independent to C by construction of the operator Ind ; therefore, $FAL(C) \cap FAL(Ind(C, \varphi)) = \emptyset$. Furthermore, $s \notin FAL(C)$.

4 Belief Revision Between Conjunctive Forms

Our belief revision method is based on the following two properties:

1. If $\forall s \in Fals(\phi)$ holds that $s \in Fals(K)$, then $K \models \phi$.
2. If $\exists s \in Fals(\phi)$, and $s \notin Fals(K)$, then $K \not\models \phi$.

The first case considers all assignments $Fals(\phi)$ are in the set $Fals(K)$, which show that $K \models \phi$. And in this case $K' = K$, since no need to change the KB K . In the second case, the sets of assignments S such that $\exists \varphi \in \phi$, $S \subseteq Fals(\varphi)$ and $S \not\subseteq Fals(K)$ are detected.

Algorithm 1 Procedure $Ind(\varphi_i, K)$

Input: K : A KB, φ_i : Clause with new information
 Push(φ_i, V); $Fs = \emptyset$; {Output in Fs a CF (Set of clauses)}
for all $C_j \in K$ **do**
 while ($V \neq \emptyset$) **do**
 $\varphi = \text{Pop}(V)$; {Try following clause}
 $Fs = Fs - \varphi$; {Remove the exit clause}
 $Nc = Ind(\phi, C_j)$; {Form: $Nc \wedge C_j \models \phi$ }
 if ($Nc \neq \emptyset$) **then**
 $Fs = Fs \cup Nc$; {Only if there are clauses to add}
 end if
 end while
 $V = Fs$; {Next iteration considers new clauses}
end for
 Return(Fs)

Example 1. Let $K = (\neg p \vee q \vee s) \wedge (\neg q \vee \neg r \vee s) \wedge (\neg q \vee r \vee \neg s) \wedge (\neg p \vee \neg q \vee r)$ and $\phi = (\neg p \vee \neg r) \wedge (\neg q \vee r) \wedge (p \vee q \vee \neg r \vee \neg s) \wedge (\neg t)$. To prove $K \models \phi$, it is equivalent to check $Fals(\phi) = \{1*1**, *10**, 0011*, ****1\} \subseteq Fals(K) = \{10*0*, *110*, *101*, 110**\}$. In each cell of column 2 of the table 1, it will show the result of $Ind(\varphi_i, C_j)$.

For example, let $C_i = (x \vee q)$, and $C_j = (\neg x \vee q)$, then $C_i \wedge C_j = (q)$. In terms of the falsifying strings clauses, we denote such a reduction as $Varcom(A_i, A_j)$. In

Table 1. Building $Ind(\phi, K)$ operator

$\phi \backslash K$	10*0*	*110*	*101*	110**	S
1*1**	111** 1011*	1111* 1011*	1111* 1011*	1111* 1011*	1111* 1011*
*10**	*10**	*10**	*100*	0100*	0100*
0011*	0011*	0011*	0011*	0011*	0011*
****1	0***1	00**1 010*1 01111	00**1 01001 01111	00**1 01001 01111	00**1 01001 01111
	11**1	110*1 11111	11001 11111	\emptyset 11111	\emptyset 11111
	10*11	10*11	10*11	10*11	10*11

the case of our example, we have: $Varcom\{1111^*, 1011^*\} = \{1^*11^*\}$. It is relevant to apply the reduction operation by complementary literals on the strings in S , for minimizing the total number of clauses. Thus, $Varcom$ and subsumed clauses are applied in order to reduce the resulting CF, from the example 1, we have:

$S = \{1^*11^*, 0100^*, 0011^*, 0^{***}1, *1111, 10^*11\}$. Writing S as CF, $Fs = (\neg p \vee \neg r \vee \neg s) \wedge (p \vee \neg q \vee r \vee s) \wedge (p \vee q \vee \neg r \vee \neg s) \wedge (p \vee \neg t) \wedge (\neg q \vee \neg r \vee \neg s \vee \neg t) \wedge (\neg p \vee q \vee \neg s \vee \neg t)$. So, the new KB $K' = K \wedge Fs$ holds $K' \models \phi$.

5 Method Properties of Belief Revision

Theorem 2. Given two clauses φ_i and C_j , it holds that $(C_j \wedge Ind(\varphi_i, C_j)) \models \varphi_i$.

Proof. If φ_i and C_j are independent clauses, then $\varphi_i = Ind(\varphi_i, C_j)$ and therefore $(C_j \wedge Ind(\varphi_i, C_j)) = (C_j \wedge \varphi_i)$. So $(C_j \wedge \varphi_i) \models \varphi_i$, propositional property by: $((p \wedge q) \supset q$ and the reflexivity of the logical inference: $\varphi_i \models \varphi_i$). If φ_i and C_j are not independent, but $Ind(\varphi_i, C_j) = \emptyset$, this implies that $Fals(\varphi_i) \subseteq Fals(C_j)$ and such $C_j \models \varphi_i$. As $C_j = (C_j \wedge Ind(\varphi_i, C_j))$, then $(C_j \wedge Ind(\varphi_i, C_j)) \models \varphi_i$. When φ_i and C_j are not independent, and $Ind(\varphi_i, C_j) \neq \emptyset$, it holds that $(C_j \wedge Ind(\varphi_i, C_j)) \equiv (C_j \wedge \varphi_i)$ by (2), fulfilling that $(C_j \wedge \varphi_i) \models \varphi_i$, propositional property by: $((p \wedge q) \supset q$, and reflexivity: $\varphi_i \models \varphi_i$. Thus, for any of the three possible outcomes of $Ind(\varphi_i, C_j)$, it is true: $(C_j \wedge Ind(\varphi_i, C_j)) \models \varphi_i$.

Corollary 1. $Fals(Ind(\varphi_i, K)) \subseteq Fals(\varphi_i)$.

Proof. $Fals(Ind(\varphi_i, C_j)) = Fals(\varphi_i) - Fals(C_j)$, to iterate over each C_j of K is satisfied that $Fals(Ind(\varphi_i, K)) = Fals(\varphi_i) - Fals(K)$. And properties between sets, it holds that $Fals(Ind(\varphi_i, K)) \subseteq Fals(\varphi_i)$.

Example 2. Let $K = (\neg p \vee q \vee r \vee s) \wedge (\neg p \vee \neg q) \wedge (\neg p \vee \neg r)$ y $\phi = (\neg p \vee \neg s \vee \neg t) \wedge (q \vee \neg r \vee \neg s \vee \neg t) \wedge (\neg p \vee q \vee r \vee \neg s \vee t) \wedge (\neg p)$, check if $K \models \phi$. It is equivalent to check whether $Fals(\phi) = \{1^{**}11, *0111, 10010, 1^{****}\} \subseteq Fals(K) =$

Table 2. Application of $Ind(\phi, K)$ operator

$\phi \backslash K$	1000*	11***	1*1**	S
1**11	1**11	10*11	10011	10011
*0111	*0111	00111	00111	00111
10010	10010	10010	10010	10010
1****	11***	\emptyset	\emptyset	\emptyset
	101**	101**	\emptyset	\emptyset
	1001*	1001*	1001*	1001*

$\{1000^*, 11^{***}, 1^*1^{**}\}$. In each cell of the table 2 is shown $Ind(\varphi_i, C_j)$. As shown in Table 2, $Ind(\varphi, K)$ is applied to a greater number of strings to those shown in Table 3, because in table 3, before to apply the independence operator, the clauses $C_i \in K$ are ordered according to the size $|lit(C_j) - lit(\varphi_i)|$. Because the number of literals C_j different with $varphi_i$ determine the number of independent clauses to be generated by $Ind(\Phi, K)$

Table 3. Calculation Ind with the $C_i \in K$ sorted

$\phi \backslash K$	1*1**	11***	1000*	S
1**11	1*011	10011	10011	10011
$\varphi_2 \backslash K$	1000*	11***	1*1**	S
*0111	00111	00111	00111	00111
$\varphi_3 \backslash K$	11***	1*1**	1000*	S
10010	10010	10010	10010	10010
$\varphi_4 \backslash K$	1*1**	11***	1000*	S
1****	1*0**	100**	1001*	1001*

Therefore, before applying operator $Ind(\varphi_i, K)$, it is appropriate to order the clauses of K according to the literals of each φ_i , as shown in Table 3. It gives us an strategy on the number of independent clauses to be generated. Also, the number of clauses to be generated is reduced via reduce clauses with complementary literals, it results in the case of example 2, $S = \{1001^*, 00111\}$, whose CF is $Fs = (\neg p \vee q \vee r \vee \neg s) \wedge (p \vee q \vee \neg r \vee \neg s \vee \neg t)$. Thus, $K' = K \wedge Fs = (\neg p \vee q \vee r \vee s) \wedge (\neg p \vee \neg q) \wedge (\neg p \vee \neg r) \wedge (\neg p \vee q \vee r \vee \neg s) \wedge (p \vee q \vee \neg r \vee \neg s \vee \neg t)$.

6 The KM Postulates

We present here the analysis of the postulates on our proposed belief revision operator $K' = K \circ \Phi = K \wedge \text{Ind}(\Phi, K)$. Consider now the KM postulates.

- **(R1)** $K \circ \phi \models \phi$.
- **(R2)** If $K \wedge \phi$ is satisfiable then $K \circ \phi \equiv K \wedge \phi$.
- **(R3)** If ϕ is satisfiable, then so is $K \circ \phi$.
- **(R4)** If $K1 \equiv K2$ and $\phi1 \equiv \phi2$, then $K1 \circ \phi1 \equiv K2 \circ \phi2$.
- **(R5)** $(K \circ \phi) \wedge \gamma \models K \circ (\phi \wedge \gamma)$.
- **(R6)** If $(K \circ \phi) \wedge \gamma$ is satisfiable then also $K \circ (\phi \wedge \gamma) \models (K \circ \phi) \wedge \gamma$.

Theorem 2 shows that our belief revision operator meets the postulate **R1**. If $K \wedge \Phi$ is satisfiable and $K \models \Phi$, then each $\varphi_i \in \Phi$ is inferred from K and therefore $\text{Ind}(\varphi_i, K) = \varphi_i, i = 1, \dots, k$. As, $K \circ \Phi = K \wedge \text{Ind}(\Phi, K) = K \wedge \Phi$, fulfilling the postulate **R2**. The **R3** postulate is fulfilled if $K \circ \phi$ is satisfiable (for **R2**). But if $\text{Fals}(K) \cup \text{Fals}(\text{Ind}(\phi, K))$ would cover the entire space assignments: 2^n , then only in this case, $K \circ \phi$ is redefined. $K \circ \phi$ is redefined to hold **R3**. And it is redefined $K \circ \Phi = ((K \wedge \text{Ind}(\phi, K)) - C_j)$, selecting the clause $C_j \in K$ with the least information (note that $|\text{SAT}(C_j)|$ is minimal on the cardinality of the set of models if $|C_j|$ is maximum in K).

R4 and **R5** postulates are satisfied because our review operator is closed on conjunctive forms. On the other hand, $\text{Fals}(K \circ (\phi \wedge \gamma)) = \text{Fals}(K \wedge S \wedge \text{Ind}(\gamma, K)) = \text{Fals}(K \wedge S) \cup \text{Fals}(\text{Ind}(\gamma, K)) = \text{Fals}(K \circ \phi) \cup \text{Fals}(\text{Ind}(\gamma, K)) \subseteq \text{Fals}(K \circ \phi) \cup \text{Fals}(\gamma)$, by Corollary 1. Thus, $\text{Fals}(K \circ (\phi \wedge \gamma)) \subseteq \text{Fals}(K \circ \phi) \cup \text{Fals}(\gamma) = \text{Fals}((K \circ \phi) \wedge \gamma)$ **R5** fulfilled. If $(K \circ \phi) \wedge \gamma$ is satisfied, then $K \circ (\phi \wedge \gamma) \models (K \circ \phi) \wedge \gamma$. As $\text{Fals}((K \circ \phi) \wedge \gamma) = \text{Fals}(K \wedge \text{Ind}(\phi, K) \wedge \gamma)$, but (γ) would equal $\text{Ind}(\gamma, K)$ iff γ is independent with each clause of K , and then only in that case, we have that $\text{Fals}(K \wedge \text{Ind}(\phi, K) \wedge \gamma) = \text{Fals}(K \wedge \text{Ind}(\phi, K) \wedge \text{Ind}(\gamma, K)) = \text{Fals}(K \circ (\phi \wedge \gamma))$ and then, the postulate **R6** is held.

7 Time-Complexity Analysis

The time function for our belief operator $K \circ \phi$ will be denoted as $T_0(|\phi|, |K|)$, and it depends mainly on the runtime operator of independence: $\text{Ind}(\phi, K)$. $\text{Ind}(\phi, K)$ is obtained from the iterative calculation of $\text{Ind}(\varphi_i, K)$.

The time complexity of $\text{Ind}(\varphi_i, K)$ process is of order $O(|K| \cdot n \cdot f(|\varphi_i|, |K|))$, where $f(|\varphi_i|, |K|)$ is an entire function, that given a clause φ_i and a CF K , determines the number of terms that will return the $\text{Ind}(\varphi_i, K)$ process. Let us now analyze the maximum possible number of clauses that can be generated through the process $\text{Ind}(\varphi_i, K)$. In some cases, $\text{Ind}(\varphi_i, K)$ can generate null sets (when $\exists C_j \in K$, such that $C_j \models \varphi_i$), but in the worst cases, the time complexity of calculating $\text{Ind}(\varphi_i, K)$ depends on the length of the sets: $S_{ij} = \{x_1, x_2, \dots, x_P\} = \text{lit}(C_j) - \text{lit}(\varphi_i), j = 1, \dots, m$.

As noted in Example 2, a fixed $\varphi_i \in \phi$, it is appropriate to order the clauses $C_j \in K$ according to the cardinality of the $S_{ij}, j = 1, \dots, m$ from lowest to highest, and eliminating the clauses that are independent with φ_i . When there is no

independent clauses with φ_i , neither $S_{ij} = \emptyset$ for $j = 1, \dots, m$, the time complexity for calculating $Ind(\varphi_i, K)$ is bounded by the number of resulting clauses. In other words, it must $|Ind(\varphi_i, K)| \leq |S_{i1}| * |S_{i2}| * \dots * |S_{im}| * Poly(n)$. Where $Poly(n)$ summarizes a polynomial time on the number of variables generated applying the operator $Ind(\varphi_i, C_j)$. Then, we can infer that the time complexity $T_0(|\phi|, |K|)$ for our belief revision operator, in the worst case, is upper bounded by $Max\{|S_{i1}| * |S_{i2}| * \dots * |S_{im}| : \forall \varphi_i \in \phi\}$, eliminating polynomial factors on n (the number of variables) and the size of the KB K and the maximum value for this upper bound is $2^{(n - \min\{|\varphi_i| : \varphi_i \in \phi\})}$. Thus, $T_0(|\phi|, |K|) \leq Max\{|S_{i1}| * |S_{i2}| * \dots * |S_{im}| : \forall \varphi_i \in \phi\} \in O(2^{(n - \min\{|\varphi_i| : \varphi_i \in \phi\})})$.

8 Conclusions

We present a new method for belief revision between CF's. As K and ϕ are CF's, the review process between K and ϕ is reduced to make the review between each $\varphi_i \in \phi$ and each $C_j \in K$, simplifying the overall problem to do $|K| * |\phi|$ subproblems of review between two clauses. A logical operator, called $Ind(\varphi_i, C_j)$, was built. $Ind(\varphi_i, C_j)$ finds the clauses whose falsifying assignments is $Fals(\phi) - Fals(K)$. The correctness of our proposal is proved, and we show that it also holds the KM postulates. We also show the complexity-time of our proposal.

References

1. Alchourron, C., Gardenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* 50, 510–530 (1985)
2. Dalal, M.: Investigations into theory of knowledge base revision. In: Proc. The Seventh National Conference on Artificial Intelligence. pp. 475–479. AAAI (1988)
3. Darwiche, A., Pearl, J.: On the logic of iterated belief revision. *Artificial Intelligence* 89, 1–29 (1997)
4. Darwiche, A.: On tractable counting of theory models and its application to truth maintenance and belief revision. *Applied Non-Classical Logics* 11, 11–34 (2001)
5. De Ita, G., Zacarias, F.: Supervised and unsupervised lexical knowledge methods for word sense disambiguation. *Computers and the Humanities* 34, 103–108 (2000)
6. Doubois, O.: Counting the number of solutions for instances of satisfiability. *Theoretical Computer Science* 81, 49–64 (1991)
7. Ellis, D.: Irredundant families of subcubes. *Mathematical Proceedings of the Cambridge Philosophical Society* 150, 257–272 (2011)
8. Katsuno, H., Mendelzon, A.O.: On the difference between updating a knowledge base and revising it. *KR'91 Cambridge, MA, USA* 1, 387–394 (1991)
9. Khardon, R., Roth, D.: Reasoning with models. *Artificial Intelligence* 87, 187–213 (1996)
10. Liberatore, P., Schaerf, M.: Belief revision and update: Complexity of model checking. *Journal of Computer and System Sciences* 62, 43–72 (2001)
11. Liberatore, P., Schaerf, M.: The complexity of model checking for belief revision and update. *Journal of Computer and Systems Sciences* 62, 43–72 (2001)

12. Nebel, B.: How hard is it to revise a belief base? *Handbook of Defeasible Reasoning and Uncertainty Management Systems* 3, 77–145 (1998)
13. Pilar Pozos Parra, W.L., Perrussel, L.: Dalal's revision without hamming distance. In: *12th Mexican International Conference on Artificial Intelligence*. pp. 41–53 (2013)
14. Satoh, K.: Nonmonotonic reasoning by minimal belief revision. *Institute for New Generation Computer Technology* 358, 157–170 (1988)

Thresholding Approach based on GPU for Facial Expression Recognition

Jesús García-Ramírez¹, J. Arturo Olvera-López¹, Ivan Olmos-Pineda¹, Georgina Flores-Becerra², Adolfo Aguilar-Rico²

¹ Benemérita Universidad Autónoma de Puebla, Faculty of Computer Science, Puebla, México
gr_jesus@outlook.com, {aolvera, iolmos}@cs.buap.mx

² Instituto Tecnológico de Puebla, Puebla, México
{kremhilda, adolforico2}@gmail.com

Abstract. Facial expression recognition and its applications are currently research topics in computer science. When analyzing face images, it is necessary to apply an algorithm for pre-processing them in order to extract features from each facial expression. In this paper related to our initial research phase, we propose variations and results from algorithms for binarizing and thresholding face expression images. In addition, we present earlier results about the parallel implementation of our proposed algorithms.

Keywords. Facial Expression Recognition, Image pre-processing, Segmentation, Thresholding.

1 Introduction

In recent years, several applications have been developed about information processing, in special those related to digital images, such as facial expression recognition; some applications in this field are detection of mental disorders, detecting whether a person is lying in an interview, detection of emotions, among others. When processing digital images, it is very common to find regions of interest (ROI), i.e. extracting from the whole image those regions to be analyzed. For extracting ROI's there exist several methods, for example those based on analysis of image textures, and some other based on locating main points related to eyebrows, nose and mouth.

The process of facial expression recognition starts with a pre-processing stage usually related to tasks such as: smoothing, scaling, converting to gray scale, among others. After the pre-processing step, a process for feature extraction is applied; first, ROI's are extracted from the whole image using techniques where the background and non-background are detected. After that, a second stage is performed, where the relevant features are extracted for expressions analysis, such as eyes, eyebrows, mouth and cheeks.

Commonly the face expression analysis is carried out over images represented in RGB (Red, Green, and Blue) color model, however, other color models can provide relevant information about border, luminance or texture. For this reason, it is im-

portant to analyze the information represented by other color spaces like HSV or YUV.

When processing digital images every pixel is taking into account and depending of the image size, this process could be computationally expensive. Due the processing of digital images requires many computation resources, in special when processing video, for example, some video devices capture 1500 fps (frames per second, i.e. 1500 images per second), then the runtime and the amount of resources are huge.

Alternatives for dealing with these problems are the parallel approaches such as the architectures provided by Graphic Processing Units (GPU) in personal computers. Initially these devices were used only for graphic processing, but nowadays they can be used for parallel programing taking advantage of the processing cores into them.

In this paper we present results about our early methodology stage for pre-processing digital images for facial expressions recognition, we present an approach for pre-processing face images based on both sequential and GPU approaches.

The paper is organized as follows: Section 2 describes the problem to solve, Section 3 show the methods and the related works of facial expression recognition, Section 4 presents the proposed methodology for threshold and pre-processing, then experiments results are shown; finally, in Section 5 conclusions and future work are discussed.

2 Problem to Solve

The human-computer interaction has as main objective, the natural interaction with humans through a hybrid computer system (hardware and software), it means that the interaction must be like humans communicate among them. The human communication with each other is commonly done via gestures, speech or written sentences. But there exists another way for communicating among humans, which is related to the facial expression involved in most of the oral communications, the facial expressions are found in 55 percent of communicated messages, 38 percent and 7 percent related to voice intonation and spoken words respectively [1].

The communication among humans is mainly characterized by the fact of that, the humans can express emotions during a conversation. On the other hand, the computers are not able neither show nor understand emotions, for this reason a natural human-computer interaction is computationally difficult.

Nowadays, the basic approach about detection of facial expressions consists in analyzing the interaction between the computer and the human by a digital capture device and then a computer system will classify the emotion, in this process the number of images to process will be grow directly according the amount of images captured per second. This process could be computationally expensive, then an alternative is to develop facial expression recognition algorithms by parallel approaches, reducing execution time and resources.

The process that several researchers follow in this kind of approaches is shown in Fig. 1. The input of the system is an image or multiple images (video), and then the pre-processing image step is followed; this process can implement image filters, bor-

der detection and thresholding methods to get the descriptive face information. The next stage of the system is the feature extraction, encoding the image information to numeric values, then the system uses a model previously constructed (by a learner or classifier) to classify person's facial expression in the image.

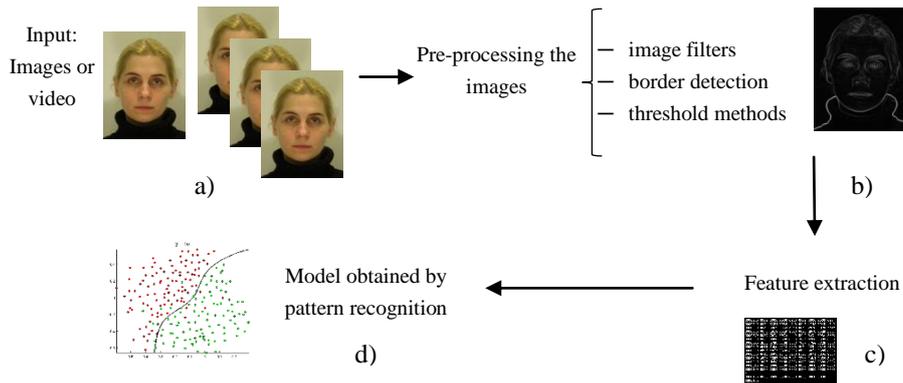


Fig. 1. Process commonly used for facial expression recognition, a) input images, face images taken from [16], b) pre-processing stage, c) feature extraction, d) model created by the classifier from information in c)

3 Related Works

The facial expression recognition is a research area since 1970 when Ekman presented two important contributions. The first one presents the six universal facial expressions: happiness, anger, sadness, disgust, surprise, and fear, several approaches about facial expression recognition find these emotions [2]. The second one is the Facial Expression Coding System (FACS), this system describes the movements in the face and also gives a time window to detect these movements. Action Units (AU), are smallest visually discriminable movements with some qualification in the FACS, this codification system specifies 9 AUs in the upper face, and 18 in the lower face. In addition, there are 14 head positions and movements, 9 eye position and movements, 5 miscellaneous actions units, 9 action descriptors, 9 gross behavior and 5 visible codes [3].

As it is shown in Fig.1, the first stage of the face expression system consists of pre-processing the images, in this process the system prepares the images to extract the principal features. Before implement pre-processing, the images can be transformed to another color spaces like reported in [4] where it is shown that other color space provide information different from that described by the RGB space. They show experiments with HSV, RGB, and YUV color models, as show in the Fig.2 the channels SV show more information about edges than RGB model about the image, and H channel do not give any relevant information of the image.

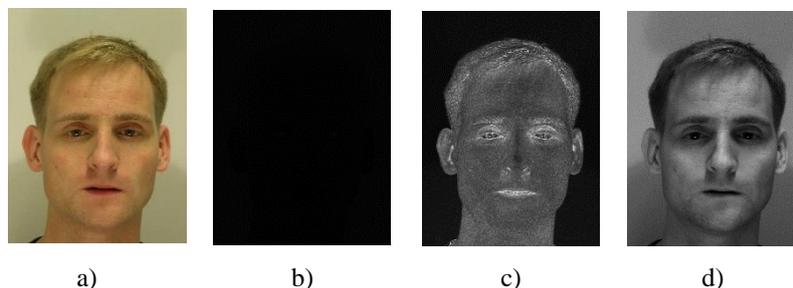


Fig. 2. a) Image represented in the RGB model b) the channel H, c) channel S d) channel V

Other pre-processing method is show in [1], this method is based on edge detection and a comparison among four border detection algorithms is reported, the algorithms compared are: Robert, Sobel, Laplace and Canny, having this last the best detection performance among the four algorithms.

Bourel F. et al. [18] report that 12 points are localized in the face of the eyebrows, nose, and mouth, then according to the distance between some of them, one of three possible states (increase, stable or decrease) is assigned for detecting the expression. Some methods pre-process the images using a threshold, which takes into account the pixels that have a higher value than a threshold. Other approaches implements models in 3D [6, 7], combine an extract Local Binary Pattern (LBP) for face localization, they use a pseudo 3D model, combining this methods to find the ROI's and extract the features. The local ROI's that several works take into account are: eyebrows, eyes, nose and mouth [8].

The feature extraction, registers the information of pre-processed images; there exist three method for information registration [10]: whole face (the face as hole entity, or register locally information), only parts of the face (like eyes, eyebrows, mouths, etc.) and edge of face (find a shape in the image). An example for registering the whole face is presented in [9], this approach pre-process the images obtaining only the face information, then the image is splited and a Gabor filter is applied, Gabor filter is used in digital images in order to analyze the information in the frequency space [19]. Other works use 3D images and find the main points of the face, then the distance between these points is registered; other approaches register the LBP from histogram of a 3D image [20, 21].

For a local registration, the register information is that related to Motion Units (MU), the MU are simply movements of face's muscles, such as movements for the eyes, eyebrows, lips, and cheeks[11].

The final stage in facial expression recognition is the classification, it uses the information from the feature extraction stage to find a model for classify, such models, are extracted by classifiers to predict or classify [22]. The classifiers use a dataset for extracting models i.e. the training set, when the dataset is previously categorized, the learning of classifier is called supervised learning, otherwise it is called unsupervised learning. Several classifiers have been applied for face expression recognition, in [11] N. Sebe et al. present a comparison among the classifiers: Naïve Bayes, Bayesian Networks, Decision trees, Nearest Neighbors. Among these classifiers, the best per-

formance was obtained by decision tree MC4. In [12, 13] an approach that implements a 3D model and a classification with Hidden Markov Models (HMM) is presented, the performance are good, but only three expressions are found (Happy, Sad, Surprise), most of the approaches implements HMM as show in [10, 14]. Other approaches like [15] use Neural Network for the classification stage, and [9] implements Nearest Neighbors classifiers.

4 Proposed Methodology and Results

In this work we present the results of the initial stage of a facial expression recognition system, we show a comparison among different technics for border detection and some variations from them. Then we propose a method to find a threshold for binarizing an image, a comparison among Isodata and Otsu threshold methods are reported. Additionally we analyze face images in different color spaces; in our experiments, we used RGB images from the MMI Database [16] and the Cohn Kanade+ Database [17].

In order to find the face and the AU regions (eyes, eyebrows, nose, cheeks, mouth), border extraction is used for detecting these regions. We applied the following border detection algorithms: mixed vertical/horizontal, gradient, Laplace, Sobel filters [1], and some variations of these.

Some obtained results are reported in Fig. 3. In order to compare these results, we computed the Structural component Coefficient (SC) which quantifies the quality (related to the original image) of the obtained image; the higher SC value the best quality in the image and $0 \leq SC \leq 1$. The expression for computing SC is described in equation (1), where $f(i, j)$ denotes the intensity value at position (i, j) in the original image whereas $f'(i, j)$ is related to the values in the resultant image.

$$SC = \frac{\sum_{j=1}^M \sum_{k=1}^N [f(j,k)]^2}{\sum_{j=1}^M \sum_{k=1}^N [f'(j,k)]^2} \quad (1)$$

The Sobel image have the highest SC value, but the image result to apply the Sobel filter contains noise that make difficult to process it, this problem is because the border transition in the original images are so faint. To solve this problem, a variation of the edge detection filters is experimented in this work; the traditional filters take into account the neighbors pixels and consider a convolutional matrix of 3x3 pixels. Then the variation we implement is to consider convolutional matrix with higher dimension depending on a value u , and then we use a matrix of $(2u+1) \times (2u+1)$ and consider only the border pixels of the convolutional matrix.

The Fig. 4 shows the performance of the border filters (vertical/horizontal, gradient, Laplace, and Sobel) using $u=5$ and their SC values, the obtained result is better than the traditional filters. The main disadvantage is that for the higher value of u the higher amount of noise in the image is obtained, but this variation results better because the borders in the image are very soft, and other issue is that for the higher u value, the thicker border in the image is obtained. Again, the Sobel filter have the highest correlation coefficient, but the amount of noise in the image is higher than in traditional filter.

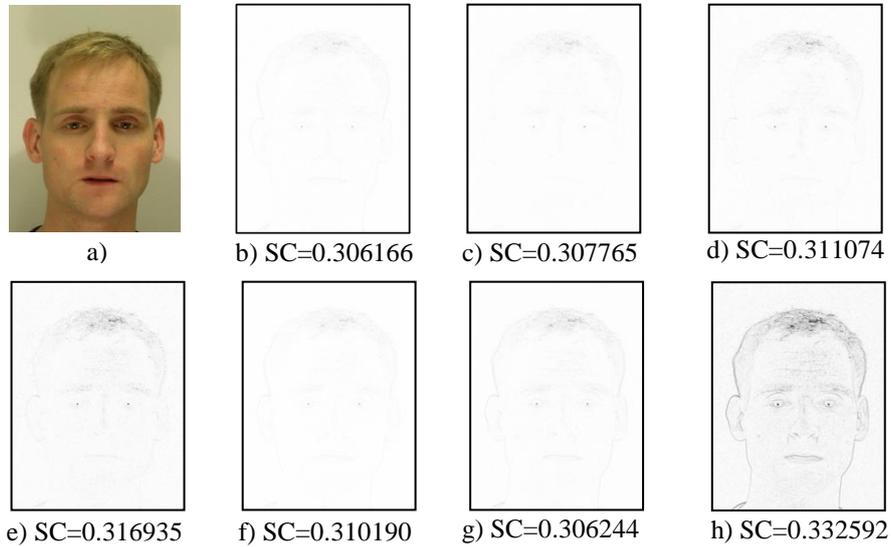


Fig. 3. Results of the border detection filter and their corresponding SC values, a) image in RGB color space, b) mixed vertical/horizontal filter, c) Laplace edge detection, d) Laplace 45 degrees, e) full Laplace filter, f) Gradient 45 degrees, g) Gradient filter, h) Sobel filter

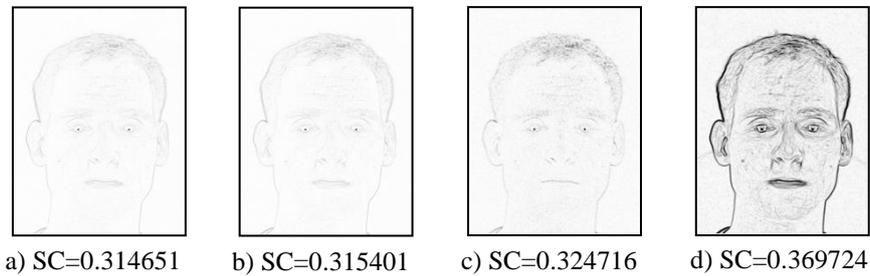


Fig. 4. Results obtained using $u=5$, a) horizontal/vertical, b) Gradient, c) Laplace, d) Sobel

To find the edge information in an image it is common binarizing in the image. This process consists of assigning a value to each pixel either 0 or 255 from a threshold. We propose a Thresholding Method based on Average (TMA), which is shown in Fig. 5. In this method, first any edge detection algorithm is applied over a gray scale image (I_B), then the frequencies histogram and the mean (Avg) from I_B are computed, the threshold t computed by TMA uses these frequencies; t corresponds to the maximum difference between two consecutives restricted to only considering frequencies higher than Avg . Mainly, t is the higher value among significant transitions.

We make a comparison against Otsu and Isodata algorithm, the results of applying the three methods is shown in Fig. 6, note that the TMA algorithm give the best edge information, less noise, and the best edge information among Otsu and Isodata.

TMA (Threshold Method based on Average)
 Input: I (Image in RGB Format), Output: t (threshold)

01. $I_G \leftarrow$ Transform I to gray scale format
02. $I_B \leftarrow$ Apply any edge detection algorithm
03. $F_{I_B} \leftarrow$ Compute the histogram of I_B
04. $avg \leftarrow$ Compute the mean of F_{I_B}
05. For $i = 0$ to 255 do
06. If $F_{I_B}(i) \leq avg$ then $F_{BI}(i) = 0$
07. $t \leftarrow \forall_{0 < i < 254} : \max\{F_{I_B}(i) - F_{I_B}(i + 1)\}$
08. Return t

Fig. 5. TMA Algorithm for finding a binarizing threshold in a RGB image

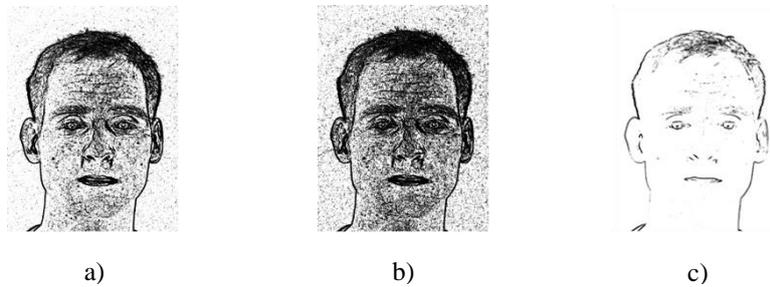


Fig. 6. Thresholding results obtained by a) Isodata, b) Otsu c) TMA algorithm

Another way of pre-processing face images consists of transforming the images into another color space. In the Fig. 2 the channels of the HSV color space are shown, then we can apply two other color spaces such as YUV and $L^*a^*b^*$. In Fig. 7 the results of converting RGB to this models are depicted, channels Y and L^* do not give relevant information, these channels show a gray scale image because they only consider information of luminance.

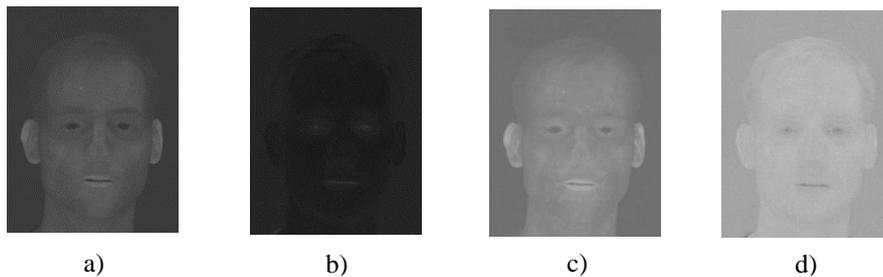


Fig. 7. Face images in different color models a) U channel of YUV color space b) the V channel of YUV color space, c) a^* channel of $L^*a^*b^*$ color model d) b^* channel of $L^*a^*b^*$

Before the system starts to implement a threshold algorithm or a border detection, the system needs to locate the face in the image. A transformation into other color space can help us for locating the face and interest points in the image, the best among

our experimental results is the b^* channel of the $L^*a^*b^*$ color space. As shown in the Fig. 8a, the marks of the eyes, mouth, nose and the border of the person in the image are sharpened. Then a border detection method like Laplace can be applied to find the border of the face and the interest points as shown in Fig. 8b.

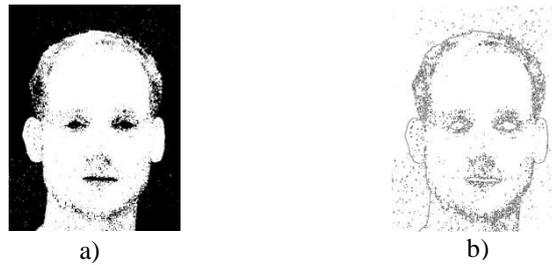


Fig. 8. Face location a) transformation of the image in figure 3a to b^* channel and the implementation of Otsu threshold algorithm, b) Laplace border detection

The previous implementation was applied over the global image (i.e. using a global binarizing threshold), but there exists other kind of implementation for only a local region in the image, this implementation can be applied over sub-regions of at most 32×32 pixels (i.e. using local binarizing threshold) and this can be implemented by a parallel approach. Applying the edge detection and the threshold methods over a local region better results can be obtained, because the noise in a region of the image are less than in the whole image, as can be seen in the Fig. 9 we applied the Otsu method over three regions of an image, the local edge detection is better than the global implementation.

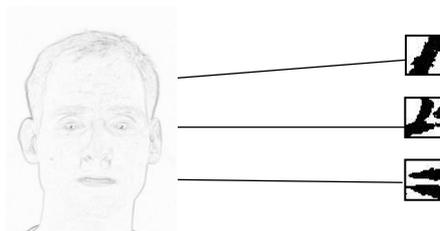


Fig. 9. Results obtained after applying local thresholding over 32×32 regions

As we mentioned before, processing face images could be expensive, in the following paragraphs, we show the results from our parallel implementation of TMA algorithm, for the parallel approach we consider splitting the image in regions and applying a local thresholding to each region in the image.

In our parallel approach we use gray scale images, we implement the TMA parallel. The implementation was applied in a GPU Nvidia with 192 cores with a clock base of 797 MHz and 2 GB in RAM, then we make test using images of different size obtained by image scaling.

In Fig. 10 we show the performance of the parallel approach TMA, note that as the higher size of the image, the higher runtime is reported, but as higher number of cores

less runtime is obtained. This experiment let us notice the usefulness of the cores processing when facing to higher dimensionality images. But if we apply this approach in a GPU with higher clock speed, or higher number of cores it can be more efficient than a sequential approach, as we mentioned the image processing is computationally expensive and more when processing video, because not only one image is processed. Then the time of processing frames of video in a sequential approach will increase.

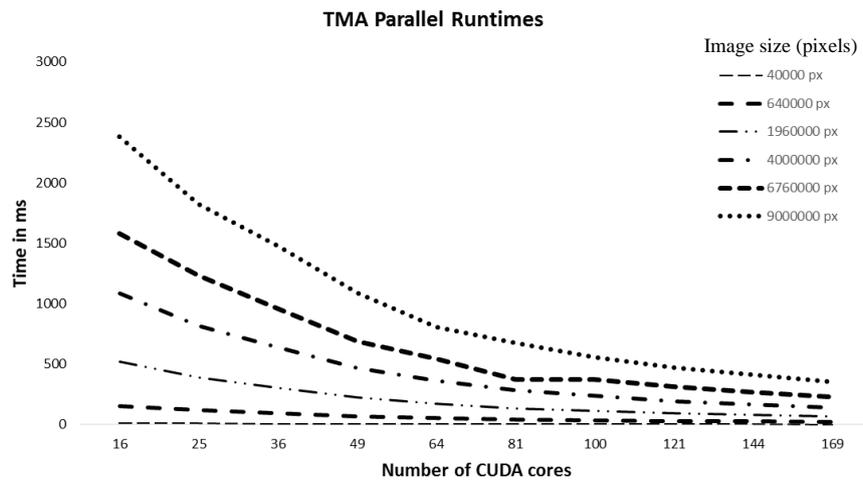


Fig. 10. Runtime of parallel approach of TMA algorithm in milliseconds with different number of cores

5 Conclusions

In order to find points of interest in facial expression recognition system, border detection and thresholding algorithms are used, we propose variations for thresholding and border detection, which have better performance than the traditional edge detection filters, this variation can be used even when the images has soft transitions in borders.

Finally, we will tested our GPU parallel implementation obtaining lower runtimes according to the increase of the number of cores, nevertheless, the load time have an impact in the global time, which makes useful the GPU approach when facing to problems requiring high sequential runtimes.

As a future work we continue in the second stage related to feature extraction from segmented images and then we will propose approaches for classifying facial expressions. In addition, we will construct a facial expression database capturing images from both visible and thermal (infrared) ranges, which will be used for testing our future approaches.

Acknowledgements

This work was partially supported by the CONACyT Mastering Scholarship 701191, the PRODEP network "Desarrollo de tecnologías de la información y comunicación" and projects VIEP-BUAP: OLLJ-ING16-I, OLPI-ING16-I.

References

1. Xiaoming C., Wushan, C.: Facial expression recognition based on edge detection. *International Journal of Computer Science & Engineering Survey*6(2), 1–9 (2015)
2. Ekman P.: Strong evidence for universals in facial expressions: a reply to Russell's mistaken critique. *Psychological Bulletin* 115(2), 268–287 (1994)
3. Cohn J., Ambadar Z., Ekman, P.: Observer-based measurement of facial expression with the Facial Action Coding System. In: J. A. Coan & J. J. B. Allen (Eds.), *The handbook of emotion elicitation and assessment*. New York, pp. 203–221. (2007)
4. Torres L., Reutter J., Lorente, L.: The importance of the color information in face recognition. In: *Proceedings of the International Conference on Image Processing, (ICIP 99)*, Vol. 3, pp. 627–631 (1999)
5. Chibelushi C., Bourel, F.: Facial Expression Recognition: A Brief Tutorial Overview. In: *On-Line Compendium of computer Vision*, Vol. 9, pp.1–5 (2002)
6. Huiquin Z., Osamu, Y.: A Fusion Approach for Facial Expression Using Local Binary Pattern and a Pseudo 3D Face Model. In: *Proceedings of International Conference on Software Engineering, Artificial Intelligence, Networking Parallel/Distributed Computing*, pp. 121–126 (2013)
7. Essa I., Pentland, A.: Coding Analysis, Interpretation, and recognition of facial expression. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 19(7), 757–763 (1997)
8. Yongqiang L., Shangfei W., Yongping Z., Qiang, J.: Simultaneous Facial Feature Tracking and Facial Expression Recognition. *IEEE Transactions on Image Processing* 22(7), 2559–2573 (2013)
9. Wenfei G., Cheng X., Venkatesh, Y., Dong H., Han L.: Facial expression recognition using radial encoding of local Gabor Features and classifier synthesis. *Pattern Recognition* 45(1), 80–91 (2012)
10. Sariyanidi E., Gunes H., Cavallaro, A.: Automatic Analysis of Facial Affect: A survey of Registration, Representation, and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(6), 1113–1133 (2015)
11. Cohen I., Sebe N., Gozman F., Cirelo M., Huang, T.: Learning bayesian network classifiers for facial expression recognition both labeled and unlabeled data. In: *Proceedings of Computer Vision and Pattern Recognition (CVPR 2003)*, pp. 595–601 (2003)
12. Le V., Tang H., and Huang, T.: Expression recognition from 3D dynamic faces using robust spatio-temporal shape features. In: *Proceedings of Face and Gesture Recognition (FG 2011)*, pp. 414–421 (2011)
13. Sandbach G., Zafeiriou S., Pantic M., Rueckert, D.: A dynamic approach to the recognition of 3D facial expressions and their temporal models. In: *Proceedings of Face and Gesture Recognition (FG 2011)*, pp. 406–413 (2011)

14. Corneau C., Ollivier M., Cohn J., Escalera, S.: Survey on RGB, 3D, Thermal, and Multimodal Approaches for Facial Expression Recognition: History, Trends, and Affect-related Applicants. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 99, 1–20 (2015)
15. Tian Y., Kanade T., Cohn, J.: Recognizing Action Units for Facial Expression Analysis. *Transactions on Pattern Analysis and Machine Intelligence* 232(2), 97–115 (2001)
16. Pantic M., Valstar M., Rademaker R., Maat, L.: Web-based database for facial expression Analysis. In: *Proceedings of International Conference on Multimedia and Expo*, pp. 5–10 (2005)
17. Kanade, T., Cohn, J., & Tian, Y.: Comprehensive database for facial expression analysis. In: *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG 2000)*, pp. 46–53 (2000)
18. Bourel F., Chibelushi C., Low, A.: Robust Facial Expression Recognition Using a State-Based Model of Spatially-Localized Facial Dynamics. In: *Proceedings of the Fifth IEEE International Conference Automatic Face and Gesture Recognition*, pp. 106–111 (2002)
19. Littlewort G., Whitehill J., Wu T., Fasel I., Frank M., Movellan J., Bartlett, M.: The computer expression recognition toolbox (CERT). In: *Proceedings of Face and Gesture Recognition (FG 2011)*, pp. 298–305 (2011)
20. Tang H., Huang, T.: 3D facial expression recognition based on automatically selected features. In: *Proceedings of the Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1–8 (2008)
21. Sandbach G., Zafeiriou S., Pantic, M.: Local normal binary patterns for 3D facial action unit detection. In: *proceedings of the International Conference of Image Processing (ICIP 2012)*, pp. 1813–1816 (2012)
22. Jiawei H., Micheline K. Jian, P.: *Data Mining, Concepts and Techniques*, Elsevier Inc., (2012)

Mathematical Modelling of the Cardiovascular System Haemodynamics

Anabel Hernández-Ramírez¹, Andrés Fraguera-Collar¹, Rafael Lemuz-López²

¹ Benemérita Universidad Autónoma de Puebla, Physical and Mathematical Sciences Department, Puebla, México

`anabel.hernandez@alumno.buap.mx`, `fraguera@fcfm.buap.mx`

² Benemérita Universidad Autónoma de Puebla, Computer Science Department, Puebla, México

`rlemuz@cs.buap.mx`

Abstract. In this paper we propose a model of the cardiovascular system, where stressed blood volume is a model parameter instead of an initial condition. Stressed blood volume (SBV) is an important indicator of fluid responsiveness, i.e., this term can help to classify patients between responders and non-responders to fluid therapy. We study a six compartment model, then using the conservation of total blood volume, one differential equation of the original model is omitted and a new model is obtained. Comparing the haemodynamic signals of the previous model and the new version, we show that the simulations are qualitatively similar. One important difference with the original model is that the initial conditions for solving the proposed model are arbitrary. This allows us to perform a sensitivity analysis using automatic differentiation of the reduced model for all model parameters without excluding any parameter a priori, unlike the authors of the original formulation have done. This analysis showed that two parameters, the heart period and the stressed blood volume, have a major effect on the performance of the model.

Keywords: Cardiovascular System, Mathematical Model, Sensitivity Analysis.

1 Introduction

In this study, a cardiovascular model of five differential equations is presented, which is a reduced proposal based on the closed-loop model of Pironet *et al.* Due to the complex circulatory interactions, the model-based method and the mathematical descriptions of the system has been used for monitoring and analyzing haemodynamic signals. Some applications of this kind of models are teaching quantitative physiology [8], simulating cardiovascular adaptation to orthostatic stress [7] and computing stressed blood volume [12].

The model proposed by Defares *et al.* is one of the first models that introduce the electrical circuit components representation of the heart haemodynamic. The cardiovascular system in the model consists of the two ventricles connected by

the systemic and pulmonary circulations. From the electrical representation of the model, the authors developed an electrical circuit analogue computer for simulating haemodynamics in real-time [2]. Furthermore, the development of cardiovascular simulators like CVSim software has been based on this model [8].

Smith *et al.* developed a minimal model of the cardiovascular system, which includes the direct ventricular interaction through modelling of the septum and the pericardium [14]. This model-based method has been validated against animal data and the subject-specific parameters can be identified by using data obtained from measurements typically available in the intensive care unit [13]. The sixteen model parameters are stressed volume, six resistances, six elastances, the cardiac period and two values for the width of the Gaussian functions which appear in the definition of the ventricles.

Considering the above model without the direct ventricular interaction and using data of pigs, Pironet *et al.* have computed the stressed blood volume, which is an important determiner of “fluid responsiveness”, i.e., this term can help to classify patients between responders and non-responders to fluid therapy [12]. In fact, Maas *et al.* have proved that in humans the capacity to improve cardiac output in response to infuse fluids and the stressed blood volume are inversely proportional under the assumption of circulatory arrest on the forearm [10].

Unlike this experimental work, Pironet *et al.* have solved a parameter identification problem for eleven of the sixteen parameters because the cardiac period, the elastance of the two ventricles and the two parameters of the width have been excluded. This problem consists in determining the parameters for which the measure of error between the model output and a corresponding set of observations is minimized. The solution of it relies on the following three steps procedure. First, are nominal values assigned to the model parameters. Second, are some parameters selected by performing a sensitive analysis on the error vector, which is associated to the parameter vector. Third, are the most sensitive parameters identified by solving a minimum norm problem with an iterative algorithm. Based on this procedure, the authors have found that stressed volume is one of the parameters to which the error vector is the most sensitive and have assigned an optimized value to it.

The aim of this study was to propose a model of the entire circulation assuming supine position based on the model of Pironet *et al.* to evaluate the relative importance of the model parameters on the performance of the model without excluding any parameter. The method used was carrying out a first order sensitivity analysis around a nominal point of the parameters, which implies to differentiate the set of state equations with respect to the vector of parameters, and to assume that the vector of input variables does not depend on the parameters. This assumption it is not accomplished by the original formulation of Pironet *et al.*, in order to avoid this, it is proposed reducing the number of differential equations by one using that there is a conserved quantity of stressed blood volume.

The paper is organized as follows: In section 2 a basic framework to the physics of cardiovascular circulation is presented. Section 3 contains a brief sum-

mary of the basic concepts for developing lumped models and the demonstration that the model of Defares *et al.* and Pironet *et al.* are equivalent. In section 4 the proposed reduction of the model of Pironet *et al.* and the sensitivity analysis of this model is presented.

2 Physics of the Cardiovascular System

The cardiovascular system is defined as a set composed of the heart, blood vessels (arteries, veins and capillaries) and blood. The set of structural and functional relations, which are established between these components, are integrated to the system to accomplish the blood circulation [6].

From a mechanical point of view, each half of the heart is composed of an atrium and a ventricle acting like a pulsatile pump coupled to the another one: the “right heart” which drives desoxygenated blood to the lungs and the “left heart” that propels oxygenated blood to all tissues of the body. In this sense, two circulations are distinguished: the *systemic circulation* and the *pulmonary circulation*. There are four heart valves which open and close to ensure a one-way flow through the heart. When the difference of pressure is positive, the valves are opened to allow the flux of blood. However, when the difference of pressure is negative, they are closed to avoid flux of blood from leaking backwards.

In haemodynamics, two important issues are the definition of the compliance chamber and the resistance vessel. The compliance chamber is considered a compartment with elastic walls, in which no opposition to blood flow is found. On the other hand, the resistance vessel compartment determines the resistance of blood flow through the “tube” . Both concepts can be defined as follows.

Definition 1. *The compliance (C) of an elastic chamber, inverse term of elastance (E), is the relation between the volume of the compartment (V) with respect to the pressure in it (P), given by the equation:*

$$V = CP + V_u = E^{-1}P + V_u, \quad (1)$$

where V_u is the unstressed volume of the chamber.

Definition 2. *The vascular resistance (R) against blood flow is defined as a relation between the pressure gradient (ΔP) and the total flux of a fluid (Q), given by the equation:*

$$Q = \frac{\Delta P}{R}, \quad (2)$$

where ΔP denotes the pressure difference of the adjacent compartments.

There are two blood volumes in which the total volume within the blood vessels can be divided. Up to 75% of the blood can be considered as unstressed volume (V_u). V_u denotes the blood which is contained in the vasculature at zero transmural pressure, i.e., the blood which does not create stress across the vessel walls because it is only filling out its natural shape. Due to the elastic properties of the

blood vessels, it is possible to add more volume, this extra volume is referred as stressed volume (V_s) since it is responsible of stretching the walls and creating a distending pressure. Using the fact, that the total blood volume in a compliance vessel is the same as the sum of unstressed volume and stressed volume in it, i.e., $V_S = V - V_U$, it is possible to re-write (1) in terms of stressed volume as it follows

$$P = \frac{1}{C} V_s = E V_s. \quad (3)$$

3 Mathematical Models of the Cardiovascular System

In this section, it is given the derivation of the equations governing the closed lumped models of Defares *et al.* and Pironet *et al.* Based on the analogies between hydraulic parameters and electrical parameters that associate pressure with voltage, volume with charge, flow with current, compliance with capacitance and vascular resistance with resistance. It is possible to represent both models as the circuit with six capacitors, six resistors and four diodes. A schematic diagram of these models is given in Figure 1. The capacitors denote the properties of compliance of systemic arteries (C_{sa}), systemic veins (C_{sv}), pulmonary arteries (C_{pa}), pulmonary veins (C_{pv}), left ventricle (C_{lv}) and right ventricle (C_{rv}). The resistors R_s and R_p are associated to the flow resistance in the systemic capillaries and pulmonary capillaries, respectively. For mimicking the heart valves, four combinations of diode-resistor are considered. Then, $R_{ao} - D_{ao}$, $R_{mi} - D_{mi}$, $R_{pu} - D_{pu}$ and $R_{tr} - D_{tr}$ represent the aortic valve, mitral valve, pulmonary valve and tricuspid valve, respectively.

For simplicity, it is assume that compliance of arteries and veins of both circulations and all the vascular resistances are constants throughout their respective compartment. On the other hand, both ventricles are considered as time-varying capacitors [16], which means that the elastance of the left ventricle, $E_{lv} : [0, \infty) \rightarrow \mathbb{R}$, and the elastance of the right ventricle, $E_{rv} : [0, \infty) \rightarrow \mathbb{R}$, are invertible functions of class C^1 , which mimic the ratio of intra-ventricular pressure and intra-ventricular volume in the respective compartment.

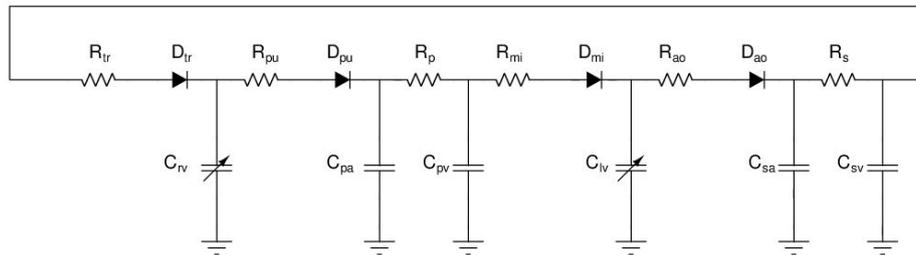


Fig. 1. Electrical analog for the cardiovascular system. Notice that $C_{lv} = E_{lv}^{-1}$ and $C_{rv} = E_{rv}^{-1}$

A first approach in studying the volume and the pressure in different compartments of the cardiovascular system considers models without control mechanisms, which means that the regulation of human blood pressure is neglected [12, 4, 14, 11, 15]. These kind of models usually assume that the function of elastance used to model ventricle muscle activation is a T -periodic function, where T denotes the heart period. In particular, the elastance functions (E_k , $k = \{lv, rv\}$) given by Smith *et al.* are considered for this work [14]. E_k is defined as:

$$E_k(t) = E_{max,k} \exp \left[-W_k \left(t \bmod T - \frac{T}{2} \right)^2 \right]$$

$E_{max,lv}$ and $E_{max,rv}$ denote the end-systolic elastance of the left and right ventricle, respectively.

In order to get the model of Defares *et al.*, which is originally given for the state vector $\mathbf{p} = [P_{lv}, P_{sa}, P_{sv}, P_{rv}, P_{pa}, P_{pv}]$. It is necessary to apply the Kirchhoff's laws to the circuit shown in Figure 1. Then, substituting the following equations.

$$\begin{aligned} Q_i &= R_i^{-1} \Delta P_i, \quad i = \{mi, ao, s, tr, pu, p\} \\ Q_j &= C_j \dot{P}_j, \quad j \in \{sa, sv, pa, pv\} \\ Q_k &= E_k^{-1} (\dot{P}_k - \dot{E}_k V_{s,k}), \quad k \in \{lv, rv\} \\ P_k &= E_k(t) V_{s,k}, \quad k \in \{lv, rv\}. \end{aligned}$$

It is gotten that the state equations for the model of Defares *et al.* are

$$\begin{aligned} \dot{P}_{lv} &= E_{lv} \left(\frac{P_{pv} - P_{lv}}{R_{mi}} H(Q_{mi}) - \frac{P_{lv} - P_{sa}}{R_{ao}} H(Q_{ao}) \right) + \frac{\dot{E}_{lv}}{E_{lv}} P_{lv}, \\ \dot{P}_{sa} &= \frac{1}{C_{sa}} \left(\frac{P_{lv} - P_{sa}}{R_{ao}} H(Q_{ao}) - \frac{P_{sa} - P_{sv}}{R_s} \right), \\ \dot{P}_{sv} &= \frac{1}{C_{sv}} \left(\frac{P_{sa} - P_{sv}}{R_s} - \frac{P_{sv} - P_{rv}}{R_{tr}} H(Q_{tr}) \right), \\ \dot{P}_{rv} &= E_{rv} \left(\frac{P_{sv} - P_{rv}}{R_{tr}} H(Q_{tr}) - \frac{P_{rv} - P_{pa}}{R_{pu}} H(Q_{pu}) \right) + \frac{\dot{E}_{rv}}{E_{rv}} P_{rv}, \\ \dot{P}_{pa} &= \frac{1}{C_{pa}} \left(\frac{P_{rv} - P_{pa}}{R_{pu}} H(Q_{pu}) - \frac{P_{pa} - P_{pv}}{R_p} \right), \\ \dot{P}_{pv} &= \frac{1}{C_{pv}} \left(\frac{P_{pa} - P_{pv}}{R_p} - \frac{P_{pv} - P_{lv}}{R_{mi}} H(Q_{mi}) \right). \end{aligned}$$

$H(t)$ is the standard Heaviside step function defined by $H(t) = 0$ for $t \leq 0$ and $H(t) = 1$ for $t > 0$, which reflects the fact that if a diode D_i is in the OFF state then the resistance in the respective compartment R_i is infinite, with $i \in \{ao, mi, pu, tr\}$. Therefore, the blood flow in the compartment is zero, i.e., $Q_i = 0$ and $H(Q_i) = 0$. Also, the fact that if a diode D_i is in the ON state then blood flow in the compartment is positive and $H(Q_i) = 1$.

Since E_{lv} and E_{rv} depends only of the time. Then, the model of Defares *et al.* can be written as the following linear system

$$\dot{\mathbf{p}} = A_D(t)\mathbf{p} \quad (4)$$

where A_D is the coefficient matrix. There is a theorem of Liapunov [1892] which affirms that the equation $\mathbf{p} = M(t)\mathbf{y}$, where M is an invertible matrix of functions of class C^1 , transforms the system (4) to $\dot{\mathbf{y}} = B(t)\mathbf{y}$, where B is a periodic coefficient matrix whose characteristic multipliers coincide with those of (4). The proof of this theorem is given in [3], Th. 2.2.7. According to the above remark, the transformation $\mathbf{p} = M(t)\mathbf{v}$, where \mathbf{v} are the state variables $\mathbf{v} = [V_{s,lv} \ V_{s,sa} \ V_{s,sv} \ V_{s,rv} \ V_{s,pa} \ V_{s,pv}]$ and the matrix M is defined as $M = \text{diag}(E_{lv}, E_{sa}, E_{sv}, E_{rv}, E_{pa}, E_{pv})$, carries the model of Defares *et al.* into the model considered by Pironet *et al.*, which is given by

$$\dot{\mathbf{v}} = A_P(t)\mathbf{v}, \quad (5)$$

where A_P is equal to $M_P^{-1}(A_D M_P - \dot{M}_P)$. Since the elastance functions E_{lv} and E_{rv} are T -periodic functions. Then both systems have the same characteristic multipliers. It is worth pointing out that the set of characteristic multipliers determine the behavior of the solutions, the existence of a periodic solution and the stability of the system.

4 Reduced Cardiovascular Model

Pironet *et al.* note that the vector of initial conditions for solving (5) must satisfy that the sum of its entries is equal to the total stressed blood volume, denoted by SBV. Then, the complete model of Pironet *et al.* is given by

$$\begin{aligned} \dot{V}_{s,lv} &= -E_{lv}V_{s,lv} \left(\frac{H(Q_{mi})}{R_{mi}} + \frac{H(Q_{ao})}{R_{ao}} \right) + V_{s,sa} \frac{H(Q_{ao})}{R_{ao}C_{sa}} + V_{s,pv} \frac{H(Q_{mi})}{R_{mi}C_{pv}}, \\ \dot{V}_{s,sa} &= E_{lv}V_{s,lv} \frac{H(Q_{ao})}{R_{ao}} - \frac{V_{s,sa}}{C_{sa}} \left(\frac{H(Q_{ao})}{R_{ao}} + \frac{1}{R_s} \right) + \frac{V_{s,sv}}{C_{sv}R_s}, \\ \dot{V}_{s,sv} &= \frac{V_{s,sa}}{C_{sa}R_s} - \frac{V_{s,sv}}{C_{sv}} \left(\frac{1}{R_s} + \frac{H(Q_{tr})}{R_{tr}} \right) + E_{rv}V_{s,rv} \frac{H(Q_{tr})}{R_{tr}}, \\ V_{s,rv} &= V_{s,sv} \frac{H(Q_{tr})}{R_{tr}C_{sv}} - E_{rv}V_{s,rv} \left(\frac{H(Q_{tr})}{R_{tr}} + \frac{H(Q_{pu})}{R_{pu}} \right) + V_{s,pa} \frac{H(Q_{pu})}{R_{pu}C_{pa}}, \\ \dot{V}_{s,pa} &= E_{rv}V_{s,rv} \frac{H(Q_{pu})}{R_{pu}} - \frac{V_{s,pa}}{C_{pa}} \left(\frac{H(Q_{pu})}{R_{pu}} + \frac{1}{R_p} \right) + \frac{V_{s,pv}}{C_{pv}R_p}, \\ \dot{V}_{s,pv} &= E_{lv}V_{s,lv} \frac{H(Q_{mi})}{R_{mi}} + \frac{V_{s,pa}}{C_{pa}R_p} - \frac{V_{s,pv}}{C_{pv}} \left(\frac{1}{R_p} + \frac{H(Q_{mi})}{R_{mi}} \right), \\ SBV &= V_{s,lv}(0) + V_{s,sa}(0) + V_{s,sv}(0) + V_{s,rv}(0) + V_{s,pa}(0) + V_{s,pv}(0). \end{aligned}$$

So the initial conditions of this model depend on the stressed blood volume. In order to get a formulation of this model in which initial conditions are arbitrary and stressed blood volume appears in the differential equations, it is important to notice that in normal conditions a constant quantity of blood (V_T) is contained in the human circulatory system. Then, assuming that V_T is known, it is feasible to compute the stressed blood volume in the systemic venous circulation $V_{s,sv}$ in terms of the other state variables. Hence,

$$V_{s,sv} = V_T - V_u - V_{s,lv} - V_{s,rv} - V_{s,sa} - V_{s,pa} - V_{s,pv} \quad (6)$$

where $V_u = V_{u,sa} + V_{u,sv} + V_{u,pa} + V_{u,pv} + V_{u,lv} + V_{u,rv}$. Finally, omitting the differential equation of $V_{s,sv}$ and substituting (6) in the remaining equations of complete model of Pironet *et al.* it is gotten the simplification of this model,

$$\begin{aligned} \dot{V}_{s,lv} &= -E_{lv}V_{s,lv} \left(\frac{H(Q_{mi})}{R_{mi}} + \frac{H(Q_{ao})}{R_{ao}} \right) + V_{s,sa} \frac{H(Q_{ao})}{R_{ao}C_{sa}} + V_{s,pv} \frac{H(Q_{mi})}{R_{mi}C_{pv}}, \\ \dot{V}_{s,sa} &= V_{s,lv} \left(E_{lv} \frac{H(Q_{ao})}{R_{ao}} - \frac{1}{C_{sv}R_s} \right) - V_{s,sa} \left(\frac{H(Q_{ao})}{C_{sa}R_{ao}} + \frac{1}{C_{sa}R_s} + \frac{1}{C_{sv}R_s} \right), \\ &\quad + \frac{1}{C_{sv}R_s} (SBV - V_{s,rv} - V_{s,pa} - V_{s,pv}), \\ \dot{V}_{s,rv} &= -V_{s,rv} \left(\frac{E_{rv}H(Q_{tr})}{R_{tr}} + \frac{E_{rv}H(Q_{pu})}{R_{pu}} + \frac{H(Q_{tr})}{R_{tr}C_{sv}} \right) + V_{s,pa} \left(\frac{H(Q_{pu})}{R_{pu}C_{pa}} \right. \\ &\quad \left. - \frac{H(Q_{tr})}{R_{tr}C_{sv}} \right) + \frac{H(Q_{tr})}{R_{tr}C_{sv}} (SBV - V_{s,lv} - V_{s,sa} - V_{s,pv}), \\ \dot{V}_{s,pa} &= E_{rv}V_{s,rv} \frac{H(Q_{pu})}{R_{pu}} - \frac{V_{s,pa}}{C_{pa}} \left(\frac{H(Q_{pu})}{R_{pu}} + \frac{1}{R_p} \right) + \frac{V_{s,pv}}{C_{pv}R_p}, \\ \dot{V}_{s,pv} &= E_{lv}V_{s,lv} \frac{H(Q_{mi})}{R_{mi}} + \frac{V_{s,pa}}{C_{pa}R_p} - \frac{V_{s,pv}}{C_{pv}} \left(\frac{1}{R_p} + \frac{H(Q_{mi})}{R_{mi}} \right). \end{aligned}$$

4.1 Sensitivity Analysis

The sensitivity functions let to study the influence of a model parameter onto the model output. However, these functions are not dimensionless quantities, reason why it is difficult to compare the sensitivity for different parameters. To avoid this, relative sensitivity is considered.

The reduced mathematical model can be re-written as

$$\frac{dx(t)}{dt} = f(t, x(t, \mathbf{p}), \mathbf{p})$$

where $x = [V_{s,lv}, V_{s,sa}, V_{s,rv}, V_{s,pa}, V_{s,pv}]$ is the state variables vector and $\mathbf{p} = [W_{lv}, W_{rv}, T, SBV, E_{max,lv}, E_{max,rv}, E_{sa}, E_{sv}, E_{pa}, E_{pv}, R_s, R_p, R_{mt}, R_{av}, R_{tc}, R_{pv}]$ is the parameter vector.

Let \mathbf{p}^0 be the parameter vector with nominal values. The relative sensitivity functions are calculated with the equation

$$S_{ij}(t) = \frac{p_j^0}{x_i(t, p^0)} \frac{\partial x_i(t, p^0)}{\partial p_j}, \quad (7)$$

where $i \in \{1, \dots, 5\}$ $j \in \{1, \dots, 16\}$.

Using Automatic Differentiation this sensitivity functions are exactly computed. In order to determine the parameters that most greatly affect the model dynamics and since the sensitivities are time-variant functions, the integral of the absolute value of relative sensitivities in the period $[0, t_f]$

$$I_{ij} = \int_0^{t_f} |S_{ij}| dt \quad (8)$$

is calculated, where t_f is the final time points, respectively. In practice, the integral is approximated with a trapezoidal method.

5 Results

Due to the numerical instability that the terms $\dot{E}_{lv}E_{lv}^{-1}$ and $\dot{E}_{rv}E_{rv}^{-1}$, which appear in the (4) model [4]. Only the models of Pironet *et al.* and the reduction of this model are simulated. There are sixteen model parameters in these models. In this work all simulations were conducted assuming pig number 2 with 31.0 kg and stressed blood volume of 990 ml, which model parameters are reported and computed by [12], details about the assigning of these nominal values are found in the original reference. The parameters related with the elastance functions are the heart period, the width of each function, the end-systolic elastance of the left and right ventricle with the following values $T = 0.474$ (s), $W_{lv} = 68.9$, $W_{rv} = 239$ (s²), $E_{max,lv} = 1.3$ and $E_{max,rv} = 1.84$ (mm Hg/ml), respectively. The resistance in the systemic capillaries, pulmonary capillaries, aortic valve, pulmonary valve, mitral valve and tricuspid valve are $R_s = 1.69$, $R_p = 0.256$, $R_{av} = 0.04$, $R_{pv} = 0.03$, $R_{mt} = 0.05$ and $R_{tc} = 0.04$ (mm Hg s/ml). Finally, the elastance in the compartments related with the arteries and veins of both circulations are $E_{sa} = 1.03$, $E_{sv} = 0.00710$, $E_{pa} = 0.699$, $E_{pv} = 0.433$ (mm Hg/ml).

The code was entirely developed and written in Octave and makes use of Martin Fink's myAD package [5], which provides a full framework for performing automatic differentiation in the Octave-environment.

Fig. 2 shows the validation of the reduced model against the original formulation by simulation. In particular, Fig. 2a shows the pressures waveforms of the systemic circulation using the complete model of Pironet *et al.* and Fig. 2b shows the same waveforms using the reduced version of this model. A mean arterial systemic pressure of 60 mm Hg or greater indicates an adequate tissue perfusion. Derived from the minimum and maximum values of the aortic pressure waveform (dashed line), it is gotten that for the above simulations this value is 102 mm Hg. The pressure-volume loops of both ventricles using the model of Pironet *et al.* (Fig. 2c) and the reduced version of this model (Fig. 2d) are also presented. The intra-ventricular pressure in both simulations is in the range of 0-50 and 0-120 (mm Hg) for the right and left ventricle, respectively. Since the reduced model has five state variables and sixteen model parameters. There are eighty relative sensitive functions S_{ij} , $i \in \{1, \dots, 5\}$ $j \in \{1, \dots, 16\}$. Due

to the extension of this paper, in Table 1 is summarized the values for the integral of the absolute value of relative sensitivities computed using (8). The first order relative sensitivity analysis around the nominal parameter vector $p=[68.9,239,0.474,990,1.3,1.84,1.03,0.0710,0.699,0.433,1.69,0.256,0.05,0.04,0.04,0.03]$ of the reduced version of the model of Pironet *et al.* showed that heart period parameter (T) is the most sensitive parameter for all the state variables (Notice that the values of I_{13} , I_{23} , I_{33} , I_{43} and I_{53} are the greatest for each column in Table 1). In second or third place appears the values of the I_{ij} related with the relative sensitivity of the state $i \in \{1, \dots, 5\}$ and the elastance of the compartment i , i.e., a very sensitive parameter for each compartment is that related with the properties of elasticity of the respective compartment. Also the stressed blood volume appears in second or third place.

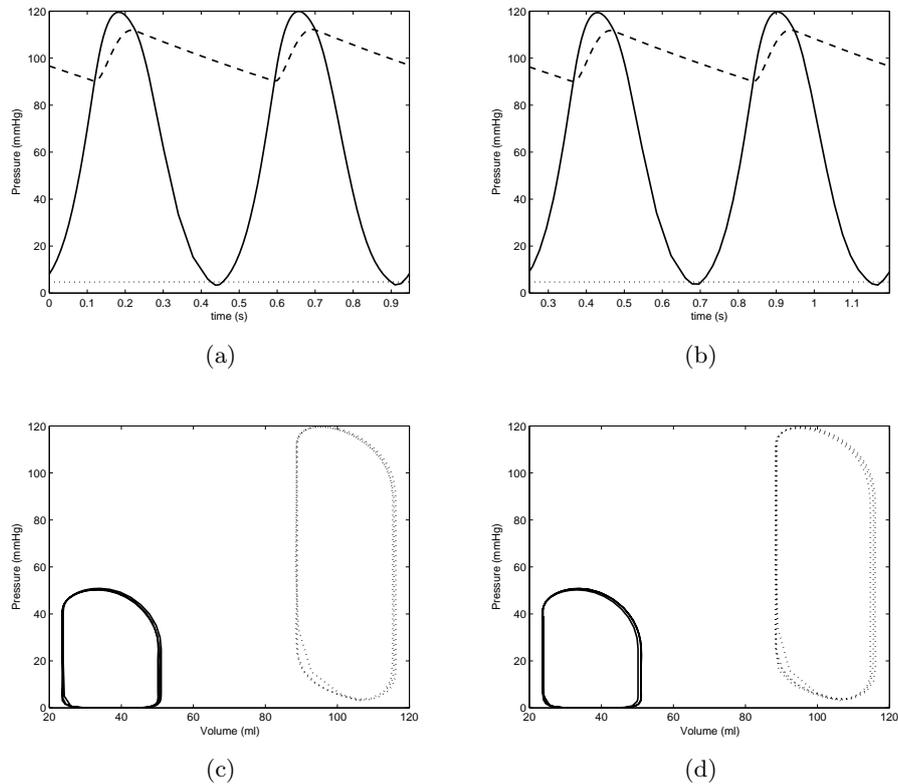


Fig. 2. Simulations of the pressures waveforms of the systemic circulation using model of Pironet *et al.* (2a) and the reduced form of this model (2b). *Solid line:* left ventricular pressure (P_{lv}), *dashed line:* aortic pressure (P_{sa}), *dotted line:* systemic vein pressure (P_{sv}). Pressure-volume loops of both ventricles using model of Pironet *et al.* (2c) and the reduced form (2d). *Solid line:* right ventricle, *dotted line:* left ventricle.

Table 1. Values for the integral of the absolute value of relative sensitivities

I_{ij}	i	1	2	3	4	5
j		V_{sa}	V_{pa}	V_{pv}	V_{lv}	V_{rv}
1	W_{lv}	3.2823	14.7651	27.5285	4.4656	7.6817
2	W_{rv}	5.1989	6.5559	6.1118	5.3763	9.0972
3	T	293.5285	576.0204	606.0967	349.3209	981.7219
4	SBV	19.3250	22.8469	21.6789	19.6447	24.1578
5	E_{lv}	2.4921	3.1688	3.6778	20.7309	2.9383
6	E_{rv}	0.4076	0.8474	0.4478	0.4043	20.2093
7	E_{sa}	22.5017	2.9541	3.2713	4.0641	2.8110
8	E_{sv}	11.9153	14.2128	13.4559	12.1294	15.0483
9	E_{pa}	1.0760	24.7842	0.9760	1.0859	3.6289
10	E_{pv}	1.0555	1.8552	22.9095	1.1959	1.3454
11	R_s	18.1934	0.7841	3.2550	13.7335	2.7022
12	R_p	1.1640	9.4063	1.3112	1.1849	5.2503
13	R_{mi}	1.3771	5.9197	11.1448	1.4349	3.0740
14	R_{ao}	0.2073	0.0968	0.2933	0.9907	0.0439
15	R_{tr}	9.7370	12.3486	11.5693	10.0734	13.0519
16	R_{pu}	0.1743	0.3028	0.2266	0.1796	1.8917

Figure 3 shows the waveform of V_{sa} using the nominal values, augmenting ten percent of the nominal value for T and augmenting ten percent of the nominal value for R_p . It is important to highlight that T is the most sensitive parameter and R_p is an insensitive parameter for V_{sa} .

6 Conclusions

It has been shown that under certain conditions the Defares *et al.* and Pironet *et al.* models are equivalent. In the sense that there is a transformation that carries system (4) into (5) and makes the set of characteristic multipliers associated to each model coincide. Moreover, it is given a reduced version of the model of Pironet *et al.*, instead of a system of six differential equations, this reduced version has five differential equations. Some differences between these models are that the first one is a linear homogeneous system of differential equations and the second one is a linear non-homogeneous system of differential equations with constant forcing terms. The initial conditions of the reduced model do not depend of any parameter, unlike the original model which depends of the stressed blood volume.

Comparing the simulations of the complete model of Pironet *et al.* and the reduced version of this model allows one to conclude that the simulations are qualitatively similar. The sensitivity analysis of the reduced model is explored. This analysis showed that two parameters, the heart period and the stressed blood volume, have a major effect on the performance of the model since the normalized ranking of the five state variables associated with these parameters have some of the highest values. Other authors have shown that beat durations

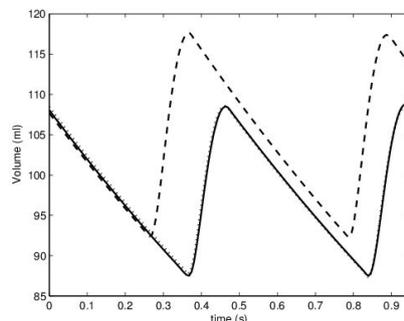


Fig. 3. Waveform for V_{sa} using *Solid line*: nominal parameters, *dashed line*: $T = 0.5214$, *dotted line*: $R_p = 0.2816$

are variable even under sinusoidal conditions causing pressure variability between beats [1, 9], the above fact seems to corroborate that heart period is a very sensitive parameter. Because the heart period is not greatly altered by the parameter identification problem according to Pironet *et al.*[12], this parameter was not considered for the sensitivity analysis of the work of Pironet *et al.* The results suggest that the reduced version can be validated. However, a rigorous proof of this fact exceeds the scope of this paper and must be addressed in a future investigation. Also as future work, the reduced model is going to be coupled with a baroreflex model, with the main objective of solving a control problem associated with the fluid therapy.

Acknowledgments

This research was facilitated through a doctoral scholarship from CONACyT under Grant 444079 and the PRODEP network of mathematical and computational modelling in medicine and demographic dynamics. We thank the reviewers for their contributions to improve this research and writing of the manuscript.

References

1. Cavalcanti, S., Belardinelli, E.: Modeling of cardiovascular variability using a differential delay equation. *IEEE Transactions on Biomedical Engineering* 43(10), 982–989 (1996)
2. Defares, J.G., Osbor, J.J., Hara, H.H.: On the theory of the cardiovascular system. *Bulletin of Mathematical Biophysics* 27, 71–83 (1965)
3. Farkas, M.: *Periodic Motions*. Springer-Verlag (1994)
4. Ferreira, A., Chen, S., Simaan, M., Boston, J., Antaki, J.: A nonlinear state-space model of a combined cardiovascular system and a rotary pump. In: *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*. pp. 897–902 (2005)

5. Fink, M.: myAD: fast automatic differentiation code in matlab (2006)
6. Guyton, A.C., Hall, J.E.: Textbook of Medical Physiology. Elsevier, eleventh edn. (2006)
7. Heldt, T., Mark, R.: Understanding post-spaceflight orthostatic intolerance - a simulation study. *Computers in Cardiology* 32, 631–634 (2005)
8. Heldt, T., Mukkamala, R., G.B., M., Mark, R.G.: Cvsim: An open-source cardiovascular simulator for teaching and research. *The Open Pacing, Electrophysiology and Therapy Journal* 3, 45–54 (2010)
9. Lerma, C., Minzoni, A., Infante, O., Jos, M.: A mathematical analysis for the cardiovascular control adaptations in chronic renal failure. *Artificial Organs* 28(4), 398–409 (2004)
10. Maas, J., Pinsky, M., Aarts, L., Jansen, J.: Bedside assessment of total systemic vascular compliance, stressed volume, and cardiac function curves in intensive care unit patients. *Anesthesia and Analgesia* 115(4), 880–887 (2012)
11. Ottesen, J.T., Olufsen, M.S., Larsen, J.K.: Applied Mathematical Models in Human Physiology. Monographs on Mathematical Modeling and Computation, SIAM (2009)
12. Pironet, A., Desaive, T., Geoffrey Chase, J., Morimont, P., Dauby, P.: Model-based computation of total stressed blood volume from a preload reduction manoeuvre. *Mathematical Biosciences* 265, 28–39 (2015)
13. Revie, J., Stevenson, D., Chase, J., Pretty, C., Lambermont, B., Ghuysen, A., Kolh, P., Shaw, G., Desaive, T.: Evaluation of a model-based hemodynamic monitoring method in a porcine study of septic shock. In: *Computational and Mathematical Methods in Medicine* (2013)
14. Smith, B., Chase, J., Nokes, R., Shaw, G., Wake, G.: Minimal haemodynamic system model including ventricular interaction and valve dynamics. *Medical Engineering and Physics* 26(2), 131–139 (2004)
15. Stergiopoulos, N., Meister, J.J., Westerhof, N.: Determinants of stroke volume and systolic and diastolic aortic pressure. *American Journal of Physiology - Heart and Circulatory Physiology* 270(6), 2050–2059 (1996)
16. Suga, H., Sagawa, K.: Mathematical interrelationship between instantaneous ventricular pressure-volume ratio and myocardial force-velocity relation. *Annals of Biomedical Engineering* 1(2), 160–181 (1972)

Impulsive Noise Removal by Adaptive Mathematical Morphology

Marisol Mares Javier¹, Carlos Guillén Galván¹, Rafael Lemuz López²

¹ Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias Físicas y Matemáticas, Puebla, México

losyram1990@gmail.com, cguillen@fcfm.buap.mx

² Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Puebla, México

rlemuz@cs.buap.mx

Abstract. In this work, an approach to the problem of impulsive noise removal in gray scale images is presented. We describe two models of adaptive mathematical morphology that improve the classical mathematical morphology in the removal of impulsive noise. The proposed method adjust the structuring elements to the local context of the image. The comparison with classical mathematical morphology denoising technique shows that the new approach better removes the salt and pepper noise, while preserves image details.

Keywords: Morphological Filtering, Opening, Closing, Composition, Structural Element.

1 Introduction

Impulsive noise (or salt and pepper), is one of the types of noise more often present in digital images. This kind of noise is caused by several factors such as, faulty memory location in hardware, problems in scanning, video sensor problems, decoding errors, transmission in noisy channel, etc. Impulsive noise elimination is a common pre-processing step before feature image analysis. A gray scale image can be described by a function f , whose domain is a subset F , in the euclidean space \mathbb{R}^2 or the discrete space \mathbb{Z}^2 , and its range is a subset \mathcal{T} of $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ or $\bar{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty, \infty\}$. A noisy image of $f : F \subseteq \mathbb{Z}^2 \rightarrow \{0, 1, \dots, L\}$, where L is the maximum value in the used intensity scale, is defined as an image $f_r : F \rightarrow \{0, 1, \dots, L\}$, affected by impulsive noise where a fraction $p \in [0, 1]$ of the points in the domain of f , which are randomly selected with uniform probability and change its values with the following expression:

$$f_r(x) = \tau(\rho(1) \leq p, \tau(\rho(1) \leq 0.5, 0, L), f(x)) \quad (1)$$

Where $\tau(c, e_v, e_f)$ is the conditional function whose arguments are the logical condition to be evaluated, the expression to be evaluated e_v when $c=1$ (true) and the expression to evaluate e_f for $c = 0$ (false). In addition, $\rho(1)$ is a function

that generates numbers with uniform distribution in the interval $(0, 1)$. Note that the values to be assigned to function f_r are the maximum and minimum in the gray scale.

In the literature impulsive noise removal methods have been purposed. One of the non-linear and robust filters most widely used is the standard median filter (SMF) [5], but as the noise level increases this filter tends to blur the image distorting its content. The Progressive Switched Median Filter (PMSF) [16], has the same drawbacks. The Adaptive Median Filter (AMF) [17], the Decision Based Algorithm (DBA) [13] and the Noise Adaptive Fuzzy Switching Median Filter for Salt-and-Pepper Noise Reduction (NAFSM) [8] are recent adaptive noise removal algorithms. Under the classical mathematical morphology framework opening and closing filters are often used for noise removal, however, even when such filters can give good results, as the level of noise degradation increases, these filters are not enough to recover good quality filtered images. Another image denoising filter commonly used is the median filter however it has the same disadvantages. In this work, an alternative impulsive noise removal filter is presented which brings satisfactory results even with a high degree of noise degradation. The proposed method preserves edges and fine feature details in the resulting images. Furthermore, experimental results shows that the proposed method has superior performance in noise removal than AMF and DBA methods when images are corrupted with high impulsive noise.

2 Noise Removal with Classic Mathematical Morphology

Classical mathematical morphology studies the images obtained after applying a number of operators which are the result of two basic operations composition, i.e., the mathematical morphology is a constructive theory [11]. These two elementary operations are defined.

Definition 1. *The erosion and dilation of function $f : F \rightarrow \mathcal{T}$ by the structural function $b : B \rightarrow \mathcal{T}$, is defined as follows,*

$$\varepsilon_b(f)(x) = f \ominus b = \bigwedge_{y \in B} \{f(x + y) - b(y)\}, \quad (2)$$

$$\delta_b(f)(x) = f \oplus b = \bigvee_{y \in B} \{f(x - y) + b(y)\}. \quad (3)$$

The function b given in the previous definition, is known as the structural function. In practice is common that the structural function b is set up as the null function, in this case the erosion and dilation definitions are reduced to the following expressions,

$$\varepsilon_B(f)(x) = f \ominus b = \bigwedge_{y \in B_x} \{f(y)\}, \quad (4)$$

$$\delta_B(f)(x) = f \oplus b = \bigvee_{y \in \hat{B}_x} \{f(y)\}. \quad (5)$$

Where \hat{B} is the symmetric set of B , $\hat{B} = \{-x : x \in B\}$, and B_x is the translation of B to the point x . The set B is called *plane structural element*. A fundamental feature of this structural element is that its size is lower than the image domain. Thus, the structural element acts as a local analyzer. Defined in this way the dilation and erosion are not the inverse the one of the other, and in general they are non invertible. This is our motivation to define other operations from their compositions.

Definition 2. *The opening and closing of f by b are defined as follows.*

$$\gamma_b(f) = f \circ b = \delta_b \varepsilon_b(f) \quad \text{and} \quad \varphi_b(f) = f \bullet b = \varepsilon_b \delta_b(f) \quad (6)$$

The derivative filters of the opening and closing operations composition are used in practice to eliminate noise in gray scale images. Figure 1 shows the result of applying these filters, using a flat structural element $B = \{(-1, 0), (0, 0), (0, 1), (1, 0), (-1, 0)\}$, to a contaminated image with impulsive noise under different levels of image degradation.



Fig. 1. From left to right, first row: An image with 20% of impulsive noise, opening morphological filter (γ_B) and closing-opening($\varphi_B(\gamma_B)$), second row: image with impulsive noise of 50%, morphological filter γ_B and $\varphi_B(\gamma_B)$, third row: image with 70% of impulsive noise, morphological filter γ_B and $\varphi_B(\gamma_B)$

As shown in Figure 1, the morphological filters open-close and close-open, are less effective as the noise degradation level increases in the image. In this paper

two models adapted to noise that slightly modify the basic operations of erosion and dilation are presented, but turn out to have a very noticeable advantage over the original operators.

3 Adaptive Mathematical Morphology

As can be seen from the definitions (2) and (3), the classical mathematical morphology studies image transformations using as structural elements a small set of pixels of fixed size and form. There is a generalization of these operations with adaptive structural elements that depend on the spatial location of the image pixels or the image content, this area is called adaptive mathematical morphology [6, 7, 9].

3.1 Impulsive Noise Removal: Model 1

A first approach to adapt the structural elements, is make them depend on its location in the space [1–4], then, once established the structural elements will remain fixed, i.e. they will not change with the input image. This approach is called “extrinsic”. Let f be a gray scaled image with domain in F , let us consider \mathcal{A} , a function that assigns to each point in the domain F , a set $\mathcal{A}(x) \subseteq F$, that can be different for every point. To define an adaptive structural element two parameters are considered, the first one is a criterion function h that brings local information on the basis of local measurements as, brightness, contrast, curvature, thickness or orientation relative to the image f , in such a way, that the size and shape of the structural element is adapted to the local image characteristics. The second parameter is the tolerance with respect to the structural element in every point, thus, the set $V_m^h(x)$ is defined, containing all the points y with $|h(y) - h(x)| \leq m$. In addition, it is needed that this set is connected by paths (path connected) with the usual topology in $F \subseteq \mathbb{R}^2$. This sets are known as Adaptive Neighborhood sets (AN sets). A fundamental feature of classical mathematical morphology is the adjunction property of erosion and dilatation, since important algebraic properties of opening and closing operators depends on this characteristic [9]. In order to ensure that our operations meet such property, the structural element must be “auto-reflected”, i.e. $\mathcal{A}(x) = \hat{\mathcal{A}}(x) = \{z \in F : x \in \mathcal{A}(z)\}$, where, $x \in \mathcal{A}(y) \Leftrightarrow y \in \mathcal{A}(x)$. The definition of erosion and dilation using adaptive structuring elements (auto-reflected AN set), are described as follows.

$$\epsilon_{\mathcal{A}}(f)(x) = \bigwedge_{y \in \mathcal{A}(x)} f(y), \quad (7)$$

$$\delta_{\mathcal{A}}(f)(x) = \bigvee_{y \in \mathcal{A}(x)} f(y). \quad (8)$$

Classic morphological filters do not satisfactorily recover the corrupted images with impulsive noise since this type of filters modifies all values in the image

(even when the pixels have no noise) using the same structural element. Thus, in the first model we propose to modify the structural element according to their position in space, to modify only the necessary image information. Filtering follows the next scheme.

1. Establish a set R , of points containing all corrupted points with impulsive noise in the image.
2. Define two adaptive structural elements, one for the elements in the set R , and another one for the complement of R , R^c .
3. Apply adaptive opening and closing compositions, using these structural elements.

Let $f : F \subseteq \mathbb{Z}^2 \rightarrow \{0, 1, 2, \dots, L\}$, the gray scale image contaminated with impulsive noise. Note that points contaminated with impulsive noise in this image are the dots whose values are 0 or L . Let R be the set containing all points where the noise function f has the value 0 or L . We define the following sets:

$$\mathcal{A}(x) = \begin{cases} B_x & \text{si } x \in R \\ \{x\} & \text{si } x \in R^c. \end{cases}$$

Where B is a fixed connex and symmetric structural element. Then, $\mathcal{A}(x)$ is an adaptive structural element. In figure 2, it is shown the result of applying the adaptive opening and closing filters to an image with varying degrees of degradation using the structural element $B = \{(-1, 0), (0, 0), (0, 1), (1, 0), (-1, 0)\}$.

3.2 Impulsive Noise Reduction: Model 2

A priori limitations imposed on the size and shape of the structural element may not be the most suitable as other operations are applied, as a result of the elementary composition. A second approach called “intrinsic” [4,7,10], generalize the morphological operators, allowing the structural elements to depend on the input image. Definitions of erosion and dilation in this approach are similar to the definitions (2) and (3), with a slight but significant change in the structural elements. These definitions are described as follows.

$$\epsilon_{\mathcal{A}^f}(f)(x) = \bigwedge_{y \in \mathcal{A}^f(x)} f(y), \tag{9}$$

$$\delta_{\mathcal{A}^f}(f)(x) = \bigvee_{y \in \mathcal{A}^f(x)} f(y). \tag{10}$$

The set $\mathcal{A}^f(x)$ is an adaptive structuring element, thus, $\mathcal{A}^f(x)$ is an auto-reflected set AN. In this case, the restriction that the structural element is auto-reflected does not ensure the adjunction property, see [9]. The opening and closing operations are defined as $\gamma_{\mathcal{A}^f}(f) = \delta_{\mathcal{A}(\epsilon_{\mathcal{A}^f}(f))}(\epsilon_{\mathcal{A}^f}(f))$ and $\varphi_{\mathcal{A}^f}(f) = \epsilon_{\mathcal{A}(\delta_{\mathcal{A}^f}(f))}(\delta_{\mathcal{A}^f}(f))$, respectively.



Fig. 2. From left to right, first row: An image with 20% of impulsive noise, opening morphological filtering ($\gamma_{\mathcal{A}}$) and opening-closing ($\varphi_{\mathcal{A}}\gamma_{\mathcal{A}}$), second row: image with 50% of impulsive noise, morphological filtering $\gamma_{\mathcal{A}}$ and $\varphi_{\mathcal{A}}\gamma_{\mathcal{A}}$, third row: image with 70% of impulsive noise, morphological filtering $\gamma_{\mathcal{A}}$ and $\varphi_{\mathcal{A}}\gamma_{\mathcal{A}}$

In this particular case, applying a first morphological operation to a noisy image we get a new image with less noise, we might consider to redefine the structural elements and streamline the process. To apply this second approach to eliminate impulsive noise, we propose the following scheme.

1. Establish a set R , of points containing all points corrupted with impulsive noise in the input image.
2. Define two adaptive structural elements, one for the elements in the set R , and another one for the complement of R , R^c .
3. Apply an erosion (dilation) according to these structural elements.
4. Define a new image, result of the previous step, as the input image and return to step 1.

Let $f : F \subseteq \mathbb{Z}^2 \rightarrow \{0, 1, 2, \dots, L\}$, the gray scale image corrupted with impulsive noise and R_f the set containing all points where the noise function f has the value 0 or L . We define the adaptive structural elements as:

$$\mathcal{A}^f(x) = \begin{cases} B_x & \text{if } x \in R_f \\ \{x\} & \text{if } x \in R_f^c. \end{cases}$$

Where B is a fix connex and symmetric structural element. Figure 3 shows the result of applying the adaptive opening and closing filters to an image with

varying degrees of degradation using structural element $B = \{(-1, 0), (0, 0), (0, 1), (1, 0), (-1, 0)\}$.



Fig. 3. From left to right, first row: image with 20% of impulsive noise, opening morphological filtering (γ_{Af}) and opening-closing ($\varphi_{\gamma_{Af}}(\gamma_{Af})$), second row: image with 50% of impulsive noise, morphological filtering γ_{Af} and $\varphi_{\gamma_{Af}}(\gamma_{Af})$, third row: image with 70% of impulsive noise, morphological filtering γ_{Af} and $\varphi_{\gamma_{Af}}(\gamma_{Af})$

4 Comparison with Other Methods

In this section the performance of the proposed method is evaluated and compared with other algorithms including SMF, PMSF, DBA, AMF y NAFSM using the Lenna image of size 510×510 degraded with different noise levels from 10% to 90% with 10% increments. The image restoration quality is measured with two metrics: Peak signal-to-noise ratio, PSNR, a metric widely used for image quality evaluation [14] and SSIM [15], a metric based on the fact that the object structures are independent of the lighting conditions. The evaluation results are presented in Table 1, and Figure 5 shows the performance of the methods for each metric. From the results we observed that for noise level lower than 30% the NAFSM, PMSF, AMF and DBA algorithms have slightly better results than the proposed methods under the used metrics, however, when noise corruption is higher that 40% , the results of the proposed algorithm are better than the PMSF, AMF and DBA algorithms, however, when the proposed

method is compared against the NAFSM algorithm, it can be noticed that the NAFSM algorithm has better performance when PSNR metric is used while our algorithm obtains better results under the SSIM metric. Figure 6 shows the resulting images from the application of the algorithms SMF, DBA, AMF, NAFSM and the proposed adaptive algorithm when images are corrupted with impulsive noise of 70%, 80% and 90%. In this figure it can be seen however, when the proposed method is compared against the NAFSM algorithm, it can be noticed that the NAFSM algorithm has better performance when PSNR metric is used while our algorithm obtains better results under the SSIM metric and that the proposed algorithm preserves better image details, edges and light than the other the DBA, AMF algorithms. Finally, in tables 2 and 3 the SMF, PMSF, DBA, AMF, NAFSM, and the proposed method are compared using the Barbara and Einstein 4 images both of size 510×510 pixels with impulsive noise in the range of 10% to 90%, here it can be observed that the proposed method is better than the PMSF, AMF, and DBA algorithms and has similar performance to the NAFSM algorithm.

In practice it was observed that the proposed algorithm converges with different number of iterations of adaptive filters. Using the structural element $B = \{(-1, 0), (0, 0), (0, 1), (1, 0), (-1, 0)\}$, for a noise level of 10% it is enough the application of an opening operator; for noise level of 20% and 30% it must be applied an opening followed by a closing; for a noise level of 40%, 50% and 60% it must be applied the composition of filters openingclosingopening; for noise level between 70% and 80% the filter composition must be closing-opening-closing opening; while for noise level of 90% the composition filter must be opening-closing-closing-opening. It was also noted that the application of more filters to the restored images does not diminish their quality.

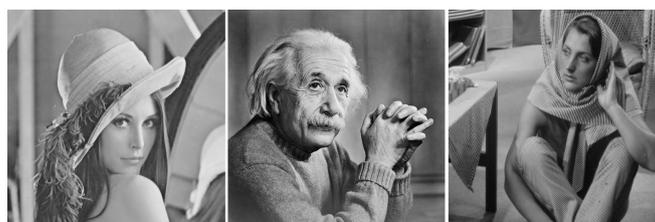


Fig. 4. From left to right: Lenna, Einstein and Barbara

5 Results

In figure 1, it can be notice that classical morphological filters are quite effective in removing noise, when noise is not as aggressive but, once the degree of degradation increases, these filters are ineffective and slow. Figure 2 shows that

Table 1. Values of PSNR in dB and SSIM for different algorithms in the Lenna image contaminated with different levels of impulsive noise

Noise %	SMF		PSMF		DBA		AMF		NAFSM		Proposed	
	PSNR	SSIM	PSNR	SSIM								
10	35.86	0.9757	37.01	0.9903	43.82	0.9991	41.64	0.9981	40.80	0.9979	38.24	0.9971
20	29.73	0.9260	32.23	0.9790	39.49	0.9971	38.36	0.9961	37.41	0.9947	35.54	0.9944
30	23.80	0.7566	28.39	0.9485	36.24	0.9936	35.83	0.9929	35.52	0.9906	34.11	0.9918
40	19.12	0.4879	24.88	0.8647	34.09	0.9873	33.77	0.9877	34.04	0.9849	33.02	0.9895
50	15.28	0.2330	15.14	0.3113	31.35	0.9745	31.71	0.9789	32.57	0.9765	31.88	0.9854
60	12.37	0.1040	12.27	0.1800	29.05	0.9546	30.05	0.9672	31.40	0.9672	30.76	0.9797
70	9.99	0.0939	9.93	0.0997	26.20	0.9128	27.96	0.9450	30.20	0.9544	29.55	0.9700
80	8.18	0.0533	8.14	0.0571	23.68	0.8461	25.95	0.9068	28.78	0.9337	28.30	0.9523
90	6.68	0.0249	6.67	0.0268	20.13	0.6762	22.95	0.8118	26.43	0.8847	26.07	0.9030

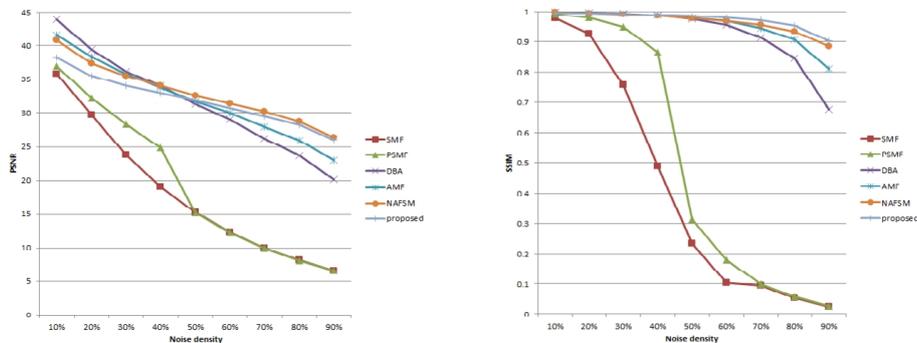


Fig. 5. Comparative graphs of PSNR in DB and SSIM for noise removal algorithms in Lenna image contaminated with different levels of impulsive noise

there is an important improvement after noise removing in the corrupted image, however the execution time of the algorithm is slow when properly parameter settings are able to obtain a satisfactory restoration. An important parameter is the size of the structural element, the noise is removed faster if a bigger structural element is used, however the image tends to get blurred losing definition in the edges, in real applications the size of the structural element must be set according to the size of the features that must remain in the input image. Another reason is that the modified points are always the same. Suppose that we have 3 points, x , y and z , horizontal aligned in the discrete domain of the image with values in the set $\{0, 1, \dots, L\}$ and, $f(x) = 0$ y $f(y) = f(z) = L$. Then, after applying an erosion with a structural element that fulfill these three points, then, the value of the function in y changes its value to L , and let us suppose that the point z is no longer noise, i.e. its value is neither 0 nor L . Let us suppose that the following operation it must be an erosion, then, since we consider again the point z to be filtered, this point remain a noisy point with value of zero. Thus,



Fig. 6. Results of noise removal algorithms for the Lenna image. From left to right: 1st column: image with impulsive noise of 70%, 80% and 90%, 2nd column: Output of MF, 3rd column: DBA, 4th column: AMF, 5th column: NAFSM, 6th column: the proposed method

Table 2. Values of PSNR in dB and SSIM for different algorithms in the Einstein image contaminated with different levels of impulsive noise

Noise %	SMF		PSMF		DBA		AMF		NAFSM		Proposed	
	PSNR	SSIM	PSNR	SSIM								
50	14.94	0.3020	20.23	0.6610	27.06	0.9287	26.59	0.9257	27.90	0.9371	26.45	0.9356
60	12.08	0.1699	12.03	0.1797	25.30	0.8892	23.34	0.8953	26.92	0.9134	25.56	0.9153
70	9.77	0.0961	9.72	0.1022	23.62	0.8342	24.10	0.8581	26.02	0.8854	24.71	0.8881
80	7.91	0.0551	7.88	0.0586	21.57	0.7492	22.61	0.7987	24.96	0.8478	23.68	0.8449
90	6.43	0.0265	6.41	0.028	19.16	0.6084	20.82	0.7049	23.44	0.7818	22.29	0.7758

the third approach shows that the elimination of noise streamlines every step if, now begin to discard these points that need no longer to be filtered.

6 Conclusions

We proposed two models of adaptive mathematical morphology for impulsive noise removal that have advantages over the classical mathematical morphology model. The results are positive in both models, as they can properly adapted to the essential characteristics of impulsive noise. The model based on the “intrinsic” approach turns out to have advantages over “extrinsic” approach, since it is adjusted in every step to the new intensities of the image to be filtered. As future work, we propose to use other structural elements to improve the results and correct the edge distortion in the images. In addition, the complexity analysis between algorithms could be evaluated.

Table 3. Values of PSNR in dB and SSIM for different algorithms in the Barbara image contaminated with different levels of impulsive noise

Noise %	SMF		PSMF		DBA		AMF		NAFSM		Proposed	
	PSNR	SSIM	PSNR	SSIM								
50	14.67	0.3012	19.38	0.603	24.78	0.8844	24.21	0.8777	25.99	0.9112	24.44	0.8977
60	11.89	0.1693	15.70	0.4070	23.44	0.8395	23.17	0.8429	25.09	0.8832	23.57	0.8715
70	9.66	0.0912	9.63	0.0978	22.19	0.7840	22.15	0.8050	24.19	0.8484	22.78	0.8380
80	7.90	0.0496	7.88	0.0527	20.60	0.6918	21.14	0.7471	23.21	0.8048	21.80	0.7899
90	6.43	0.0266	6.42	0.0278	18.15	0.5215	19.54	0.6431	21.88	0.7291	20.72	0.7164

Acknowledgments

We thank to CONACyT, PRODEP and VIEP-BUAP for the financial support. Also, we thank the reviewers for their contributions to improve this research and writing of the manuscript.

References

1. Angulo, J.: Morphological Bilateral Filtering and Spatially-Variant Adaptive Structuring Functions. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 212–223. Springer, Heidelberg (2011)
2. Bouaynaya, N., Schonfeld, D.: Spatially variant morphological image processing: theory and applications. In: Proceedings of SPIE, vol. 6077, pp. 673–684 (2006).
3. Bouaynaya, N., Schonfeld, D.: Theoretical foundations of spatially-variant mathematical morphology part II: Gray-level images. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(5), 837–850 (2008)
4. Debayle, J., Pinoli, J.: Spatially Adaptive Morphological Image Filtering using Intrinsic Structuring Elements. Image Analysis and Stereology 24(3), 145–158 (2005)
5. Huang, T., Yang G., and Tang G.: A fast two-dimensional median filtering algorithm. IEEE Trans. Acoust., Speech, Signal Processing 27(1), 13–18 (1979)
6. Heijmans H.J., and Ronse C.: The Algebraic Basis of Mathematical Morphology I, Dilations and Erosions. Comput. Vision Graphics Image Processing 50(3), 245–295 (1990)
7. Maragos, P.A., Vachier, C.: Overview of adaptive morphology: Trends and perspectives. In: Proc. of IEEE International Conference on Image Processing, pp. 2241–2244 (2009)
8. Vin Toh, K.K. and Mat Isa, N.A.: Noise Adaptive Fuzzy Switching Median Filter for Salt-and-Pepper Noise Reduction. IEEE Signal Processing Letters 17(3), 281–284 (2010)
9. Roerdink, J.B.T.M.: Adaptivity and group invariance in mathematical morphology. In: 16th IEEE International Conference on Image Processing (ICIP), pp. 2253–2256 (2009)
10. Salembier, P.: Study on nonlocal morphology operators. In: 16th IEEE International Conference on Image Processing (ICIP), pp. 2269–2272 (2009)
11. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, London, United Kingdom, (1982)

12. Soille, P.: *Morphological Image Analysis: Principles and Applications*, 2nd Ed., Springer-Verlag, Berlin Heidelberg, Germany, (2003)
13. Srinivasan, K.S., and Ebenezer, D.: A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises. *IEEE Signal Processing Letter* 14(3), 189–192 (2007)
14. Wang, Z., Bovik, A.C.: Mean squared error: love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine* 26(1), 98–117 (2009)
15. Wang Z., Bovik, A.C., Sheikh H.R., and Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4), 600–612 (2004)
16. Wang, Z. and Zhang, D.: Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images. *IEEE Trans. on Cir. and Sys* 46(1), 78–80 (1999)
17. Zhao, Y., Li D., Li Z.: Performance enhancement and analysis of an adaptive median filter. In: *International Conference on Communications and Networking*, pp. 651–653, (2007)

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
junio de 2016
Printing 500 / Edición 500 ejemplares

