

RESEARCH ON COMPUTING SCIENCE

ISSN: 1665-9899

Advances in Computer Science in Mexico



**Alexander Gelbukh
Hiram Calvo
(Eds.)**

Vol. 13



Advances in Computer Science in Mexico

**Avances en la ciencia de la computación
en México**

Research on Computing Science

Series Editorial Board

Comité Editorial de la Serie

Chief Editors:

Editores en Jefe

Juan Luis Díaz de León S. (Mexico)

Gerhard Ritter (USA)

Jean Serra (France)

Ulises Cortés (Spain)

Associate Editors:

Editores Asociados

Jesús Angulo (France)

Oscar Camacho (Mexico)

Jihad El-Sana (Israel)

Jesús Figueroa (Mexico)

Alexander Gelbukh (Russia)

Ioannis Kakadiaris (USA)

Serguei Levachkine (Russia)

Petros Maragos (Greece)

Julian Padget (UK)

Miguel Torres (Mexico)

Mateo Valero (Spain)

Cornelio Yáñez (Mexico)

Editorial Coordination:

Coordinación Editorial

José Angel Cu Tinoco

Miguel R. Silva Millán

Format:

Formación

Sulema Torres Ramos

Research on 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 13, mayo, 2005. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de Licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor Responsable: *Juan Luis Díaz de León Santiago, DISJ 690604*

Research on Computing Science is published by the Centre for Computing Research of IPN. Volume 13, May, 2005. 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, May, 2005, in the IPN Graphic Workshop – Publication Office.

Volume 13

Volume 13

Advances in Computer Science in Mexico

Avances en la ciencia de la computación
en México

Volume Editors:

Editores del Volumen

Alexander Gelbukh

Hiram Calvo

Instituto Politécnico Nacional
Centro de Investigación
en Computación
México, 2005



ISSN: 1665-9899

Copyright © 2005, Instituto Politécnico Nacional

Editores

Alexander Gelbukh, Hiram Calvo

Instituto Politécnico Nacional (IPN)

Centro de Investigación en Computación (CIC)

Col. Zacatenco, CP. 07738, México, D.F., México.

E-mail: gelbukh@gelbukh.com, likufanele@likufanele.com

<http://www.gelbukh.com>

Published by the Instituto Politécnico Nacional

Centro de Investigación en Computación

Av. Juan Dios Bátiz s/n esq. M. Othón de Mendizábal

Unidad Profesional "Adolfo López Mateos", Zacatenco

07738, México DF, Mexico

<http://www.ipn.mx>

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

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

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, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of 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.

Printing: 500

Printed in Mexico

Tiraje: 500

Impreso en México

Prólogo

La ciencia de la computación tiene un papel cada vez más preponderante en el quehacer diario humano. Cada día, de manera directa o indirecta, nos encontramos interactuando con el resultado de investigaciones en esta ciencia, relativamente nueva, en nuestra vida diaria. Orgullosamente podemos decir, que nuestro país no se encuentra excluido de este fenómeno mundial. En México, el papel esencial de la computación se entiende completamente a todos los niveles, desde el ambicioso programa de modernización computacional del país llamado e-México hasta el apoyo gubernamental al desarrollo e implantación de nuevo software y la educación técnica en la computación y las áreas afines.

El Instituto Politécnico Nacional está a la vanguardia de este esfuerzo nacional e internacional, tanto en la generación de nuevo conocimiento y nuevas aplicaciones de la computación como en la preparación de los recursos humanos de la más alta calidad en esta rama de ciencia e ingeniería y en la creación y el uso de los sistemas basados en las tecnologías computacionales para la modernización de su propio modelo educativo.

Esta importante actividad de nuestro Instituto se realiza en primer lugar a través de su moderno Centro de Investigación en Computación, con sus amplios enlaces nacionales e internacionales, con sus numerosos proyectos de investigación y desarrollo a escala nacional e internacional, con sus grandes congresos internacionales que abarcan todas las áreas más importantes de la computación, con más de mil artículos científicos publicados por sus investigadores y estudiantes en los siete años de su existencia, con varios miembros de la Academia Mexicana de Ciencias e Investigadores Nacionales de los más altos niveles entre sus profesores.

Esta obra continúa la serie de libros sobre computación preparados por especialistas de este Centro. Su profundo dominio del tema ha garantizado una alta calidad de los artículos científicos incluidos en este libro, para el máximo provecho del lector.

El libro presenta al lector los avances en las áreas tan importantes de la computación como robótica, visión, procesamiento del lenguaje natural, algoritmos, tecnologías de software, bases de datos, hardware y control. Este libro será de gran utilidad para estudiantes y profesionales de la ciencia de la computación, áreas afines así como para los especialistas en otras áreas, gerentes, empresarios y promotores de la educación que buscan nuevas formas de modernizar sus actividades a través de las aplicaciones y herramientas computacionales.

Es muy grato para mí presentar al lector estos avances de la ciencia de la computación en México, desarrollados con el ímpetu y la fuerza de nuestros jóvenes investigadores, consciente del esfuerzo realizado por todos ellos, con ahínco, fuerza y esmero. ¡Enhorabuena!

Dr. Juan Luis Díaz de León Santiago
Director del Centro de Investigación en Computación, IPN

Prefacio

El dinamismo de nuestro mundo actual exige que grandes cantidades de información sean procesadas a una velocidad que supera la capacidad de respuesta del individuo. Nuestra sociedad, industria, ciencia y educación están fuertemente vinculadas a esta vertiginosa lluvia de datos que en los días recientes sólo puede ser afrontada de con ayuda de una fuerte infraestructura tecnológica y social que ha dado lugar a la así llamada sociedad de la información.

Una parte medular de la sociedad de la información está constituida por la ciencia de la computación. Respondiendo a la demanda creciente de desarrollos en esta área, hemos seleccionado para su publicación en este volumen 22 artículos de avances de investigaciones y proyectos de 52 autores, principalmente de nuestro país. Presentamos los últimos desarrollos en una gran variedad de áreas relacionadas a la computación. Los artículos presentados en este volumen caen dentro de las siguientes categorías temáticas:

- Robótica y visión
- Procesamiento de lenguaje natural
- Algoritmos
- Tecnologías de software y bases de datos
- Hardware y control

La selección de los artículos para ser publicados en este volumen está basada en revisiones rigurosas de un equipo de expertos en ciencia computacional e ingeniería de la computación.

Este volumen es el resultado del trabajo de mucha gente. El esfuerzo editorial resultante en este volumen fue apoyado por el gobierno de México (IPN y CONACyT), la ACM (Association for Computing Machinery), CyT Americas (Fundación para la ciencia y la tecnología de las Américas), y RITOS-2. A todos ellos agradecemos su valiosa ayuda.

También deseamos agradecer a los revisores por el trabajo difícil de seleccionar los mejores artículos y ayudar a los autores para hacerlos aún mejores. Finalmente, agradecemos a los autores por su gran esfuerzo en obtener laboriosamente y describir cuidadosamente los resultados científicos presentados en este volumen, y por hacer pacientemente todos los cambios requeridos por nuestros revisores.

Mayo 2005

Alexander Gelbukh
Hiram Calvo

Volume Editorial Board

Comité editorial del volumen

Igor Bolshakov (CIC, IPN, México)
 Nieves Brisaboa (U. da Coruña, España)
 Erick Cantú Paz (CASC, USA)
 Ernesto Cuadros-Vargas (U. Católica de Santa María, Perú)
 Sofia Natalia Galicia Haro (F. Ciencias, UNAM, México)
 Alexander Gelbukh (CIC-IPN, México)
 Julio Gonzalo (UNED, España)
 San-Yong Han (U. Chung-Ang, Corea)
 Eugene Levner (Holon Academic Institute of Technology, Israel)
 Manuel Mejía Lavalle (IIE, México)
 Ruslan Mitkov (U. of Wolverhampton, Reino Unido)
 Daniel Panario (Carleton University, Canada)
 Luis Alberto Pineda Cortés (UNAM, México)
 Fuji Ren (Tokushima City U., Japan)
 Paolo Rosso (U. Politécnica de Valencia, España)
 Isaac D. Scherson (U. of California, Irvine, EEUU)
 Grigori Sidorov (CIC-IPN, México)
 Ralf Steinberger (Joint Research Centre, Italia)
 Andrei Tchernykh (CICESE, México)
 Carlos Martín Vide (U. Rovira i Virgili, España)

Collaborators

Colaboradores

Noe Alejandro Castro Sánchez
 Sulema Torres Ramos
 José Ángel Vera Félix

Contents

Contenido

Robotics and Vision

Robótica y visión

Simultaneous Planning: A Real-Time Planning Method	3
<i>Ramon F. Brena, Emmanuel Martinez</i>	
Algoritmo de navegación de móvil robótico de trabajo en sociedad para mapeo de escenarios	13
<i>Jesús Yaljá Montiel Pérez, Rodolfo Romero Herrera, Félix Serrano Talamantes</i>	
New Motion Prediction Algorithm Invariant to Rotation and Occlusion	23
<i>José Tomás Tello, Oleg Starostenko, Gennadiy N. Burlak</i>	
Filtro para la supresión de ruido impulsivo y multiplicativo para aplicaciones en imágenes médicas.....	35
<i>José Luis Varela-Benítez, Francisco Gallegos-Funes, Volodymyr Ponomaryov</i>	
Filtro mediano con peso central iterativo en el dominio Wavelet utilizando el método de redundancia en las aproximaciones	41
<i>Martínez Valdes Jesús, Gallegos Funes Francisco Javier, Acevedo Mosqueda Marco Antonio</i>	
Sistema de acceso al personal mediante la autenticación del iris	51
<i>Juan José Torres Manriquez, Edmundo René Durán Camarillo, Carlos Alberto González Dueñas, Aranzazú del Carmen González Montiel, Verónica Martínez Mora, Carlos Vidal Mendoza González</i>	

Natural Language Processing

Procesamiento de lenguaje natural

Agentes móviles para extractos de documentos	63
<i>Lorena Leal Bando, Darnes Vilariño Ayala, Fabiola López y López</i>	
Una nueva ponderación para el modelo de espacio vectorial de recuperación de información.....	75
<i>Rubí J. Cabrera, David Pinto, Darnes Vilariño, Héctor Jiménez-Salazar</i>	
El párrafo virtual en la generación de extractos.....	83
<i>Claudia Bueno-Tecpanecatl, David Pinto, Héctor Jiménez-Salazar</i>	
Un lenguaje de apoyo para tareas de procesamiento de textos	91
<i>Erika Hernández, David Pinto, Héctor Jiménez-Salazar, Jesús Lavalle</i>	

Algorithms

Algoritmos

Genetic Algorithm to Design Artificial Neural Networks in Time Series Forecasting with Intrinsic Mode Functions..... 101
V. Landassuri-Moreno, J. Figueroa-Nazuno

Método y algoritmo de árbol de cubos y sus aplicaciones para resolver diferentes problemas de optimización discreta 111
Vladimir Khachaturov, José Crispin Zavala Díaz

Semejanza entre objetos tridimensionales aplicando redes neuronales Fuzzy ART 127
A. Angeles-Yreta, J. Figueroa-Nazuno

Software Technologies and Databases

Tecnologías de software y bases de datos

UML Model of the IEEE 802.11 Privacy Service using AES-CCM (Advanced Encryption Standard, Counter-Mode/CBC-MAC) and AES Implementation 137
Sugehi Marina Merino Higareda, Miguel Ángel León Chávez

UML Model and Implementation of the IEEE 802.11 Authentication and Privacy Services using Elliptic Curves Cryptosystems..... 149
Rocío A. Aldeco Pérez, Miguel A. León Chávez

Construcción de agentes para el TAC SCM Game 2005 159
Daniel Macías Galindo, Fabiola López y López, Darnes Vilariño Ayala

Managed Web Services using WS-Manageability..... 171
Giner Alor Hernández, José Oscar Olmedo Aguirre

Using Previews to Reduce the Cost of Video-on-Demand Services 179
Jehan-François Pâris

Extensiones gramaticales del algebra relacional para el soporte de integridad en sistemas relacionales 191
Beatriz Bernábe Loranca, Ma. del Rocío Boone Rojas, Ramiro López Sales

Manejo de información cartográfica mediante base de datos espaciales distribuidas..... 203
G. Guzmán, M. Torres, R. Quintero, M. Moreno

Hardware and Control

Hardware y control

Diseño y construcción de un sistema telemétrico de registro extracelular multiunitario..... 213
Francisco J. Heredia López, Br. Zoemy Esquivel Loría

Several Control Strategies for Parallel Connected Dual Induction Engines by means of a Digital-Based System Control	i 227
<i>R. Peña-Eguiluz, M. Pietrzak-David, B. de Fornel</i>	
Author Index	241
Índice de autores	

Robotics and Vision

Robótica y visión

Simultaneous Planning: A Real-Time Planning Method

Ramon F. Brena, Emmanuel Martinez

Center for Intelligent Systems
Monterrey Institute of Technology
`ramon.brena@itesm.mx`, `A00588294@itesm.mx`

Abstract. We present a novel real-time planning method as we have applied it to the Robocup competition. The basic idea in this method is to simultaneously keep in consideration more than one plan by evaluating and maximizing an expected utility and feasibility of the current and the future possible actions. This could be better than maximizing the expected utility of just one plan or the current possible actions, because in a highly dynamic environment like Robocup, plans get frequently stuck due to unexpected events, like an unforeseen blocking, etc. In these cases, a “second-best” plan could rescue the situation. We propose a specific architecture for implementing this method in the context of a player agent in the Robocup competition. In this paper, after motivating and presenting the method, we propose a specific architecture for implementing this method in the context of a player agent in the Robocup competition and show how we have applied it to the “Borregos” Robocup team, that participated at the Robocup 2004 tournament at Lisbon. Some preliminary figures about performance and experimental evidence showing the potential of our method are presented as well.

1 Introduction

Real-time soccer is a complex game, involving a great deal of coordination between team members, and the development of strategies leading to marking as many goals as possible and receiving as few goals as possible.

Robocup simulated soccer represents that complexity to a certain degree, meaning that many interesting soccer complexities are present in the simulated soccer as provided by the soccerserver system [3], like limited vision and hearing, noise, etc.

In simulated Robocup each player is a decision-maker under uncertainty, because its vision is limited to a “cone”, so the part of the field not seen is basically unknown, and players try to overcome this limitation by using their memory, extrapolating the previous situation, using communication to exploit information available to their partners, etc., but a great amount of uncertainty remains anyway.

Thus, simulated soccer is a challenging problem from the standpoint of coordination, communication, uncertainty handling, learning, planning, and so on [8].

Many Robocup teams have applied techniques issued from the Artificial Intelligence (AI) field [14] to develop sophisticated skills. Among those AI-related techniques we find Neural Nets, Reinforcement Learning and Probabilistic Reasoning for low-level skills [15], and decision trees [16], Reinforcement Learning [17] and Multiagent Coordination methods like Coordination Graphs [11], for high-level skills.

One area that perhaps has not reached a high degree of development in Robocup is planning [1]. Due to the very dynamic nature of simulated soccer, long-term planning is pointless, as almost any long plan will fail when facing unexpected conditions. Thus, most teams are highly reactive and rely more on very polished low-level skills like ball interception and shooting than on clever playing ideas. Very few teams have applied a planning method to Robocup; in [13], a planning method using opponent modelling is applied for the Coach league team from Carnegie Mellon University. Using bayesian networks, the coach models the opponent's behavior and generates a plan. Finally, the coach communicates the plan to the players. In [5] players have different plans in their memory and they search for one plan to arrive to the opponent goal; their plans consist of dribbles and passes. The cited works use traditional planning methods where the starting point and the ending point of the plan are defined before its execution starts.

In most planning methods [18], when a plan is interrupted at execution time due to unexpected events, players have to start over from scratch a new plan -which will be most probably interrupted as well.

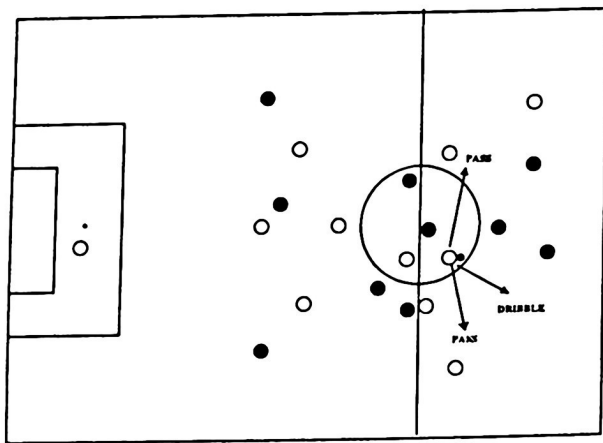


Fig. 1. Example of the problem of decision making in Robocup

Let us examine an example. Fig. 1 shows a game situation where the agent with the ball must take a decision between giving the ball to a well-positioned, but lonely partner, or filtering a long pass to the right, or even to continue dribbling, as indicated by the arrows in the figure.

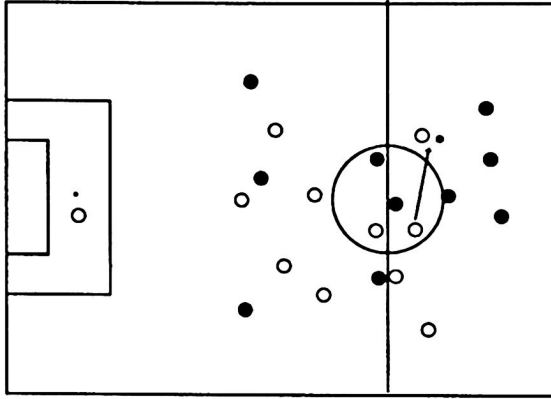


Fig. 2. Wasted action

Fig.2 shows that the after taking the decision of passing to the left, the lonely player gets stuck by enemies. Of course things could go bad as well if another action, like dribbling, is chosen, but the difference is that if the player continues dribbling, it still has the other options. The point here is not that dribbling is better than passing, but that under uncertainty is good to have options for cases where what seemed to be the best option just fails and has to be dropped. In our method we stress the importance of having as many *options* as possible.

The basic idea in our planning method is to take into consideration, when evaluating a particular possible move, how many and how good *options* the action keeps open after executing it. The advantage of doing this, instead of just selecting the best possible plan or action at a given moment according to some metric, is that in a highly dynamic environment like Robocup, plans often get stuck due to unexpected events, so the agent has to replan from scratch. In our system, for the contrary, when a plan gets stuck, it is more probable that there is another remaining option—precisely because the planning method has been fostering good options all the time.

Our planning method belongs to the class of on-line planning [7]. On-line planning interleaves planning and execution whereas in traditional planning a plan is made and then executed. Agent-centered search [9] is a technique that implements on-line planning, restricting planning to the part of the domain where the current state of the agent is found. Agent-centered search decides on the local search space, searches an action, and executes it. This process is repeated until the goal state is reached.

In the next section we will detail our planning method. In Section 3 we will describe the Robocup application of our method in our “Borregos” team, and finally in Section 4 we will discuss pros and cons of our planning method, list some results of the team, and draw some future work lines.

2 Method Description

Our planning method considers a collection of current possible actions and the next cycle future actions.¹ The collection of current possible actions (CPA) is taken from a “playbook” [2]. CPAs are specified at the *role* level, involving abstract roles like the “passer”, the “receiver”, and so on. But to be evaluated, specific players are bound to the current possible actions roles present in the current situation, so CPAs are *instantiated* to fit the specific players and positions.

The playbook have the following plays: Pass, Filtered Pass, Dribble, Outplay (a very long dribble), Clearball (a kick to certain position), and Shoot to goal. We developed some classes to manage the actions but at the end, the actions are parsed to high level actions of the UvA Trilearn code.

Now we describe the basic steps involved in our system’s operation:

- First we construct a two-level *play tree*, which is a subtree of a search tree, pruning off some not plausible plays, based on heuristics, like avoiding passes to players not in the passer neighborhood. In particular we check the compatibility between the plays in the first and second levels. These future actions options have to be “compatible” in the sense that the future actions can be performed after doing a current possible action.²
- The best evaluated current possible action is chosen from the CPA with its list of future actions. This seems entirely like conventional planning, but here the trick is that we evaluate first-level plays not entirely based on the maximum-value child, but mainly based on the accumulated value of its children, giving thus higher evaluations to plays followed by many options. The exact formula for play evaluation is presented below.
- The selected first-level action is refined through an *optimization* process. Here several variations of the action are generated, for instance changing slightly the direction, speed, etc., in order to choose the exact point yielding a maximum utility with respect to its list of future actions. This is the final decision for the agent.

Our evaluation of possible plays is based on their *expected utilities*, i.e. the product of their benefit (in case they are successful) by their probability of success; this is why we call it *combined evaluation*:

$$e(a) = bf(a) * pf(a) \quad (1)$$

where *bf* is the *evaluation of benefit* function (basically a heuristic taking into account the position of ball and players, see [12]), and *pf* the *feasibility* function,

¹ We decided not to consider more than one future cycle, so our planning trees are just two levels deep.

² For instance, after a “clear ball” you cannot make a pass, so these moves are not compatible.

which returns a number between 0 and 1. The feasibility function is supposed to correspond to the fraction of times a given play could be successful in the given situation.

The formula used to calculate the combined evaluation $E(a)$ of a first-level action a is as follows:

$$E(a) = k_1 e(a) + k_2 \max\{e(s_i(a))\} + k_3 \sum\{e(s_i(a))\} \quad (2)$$

where $s_i(a)$ the i -th successor of action a in the tree, $e(a)$ is the combined evaluation of current possible action a with respect to its feasibility and utility, $e(s_i(a))$ are the evaluations with respect to the feasibility and utility of the future actions $s_i(a)$ of the current possible action a .

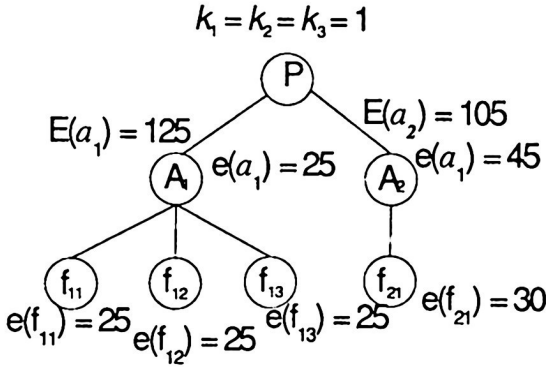


Fig. 3. A simple two-level tree

We illustrate the application of this formula to a simple tree presented in Figure 3. In conventional maximum-driven tree evaluation, the branch to the right would be selected, because this one gives a maximum value. But applying our formula, the evaluations are different. According to our formula the evaluation of node A_1 is:

$$\begin{aligned}
 E(A_1) &= e(A_1) + \max\{e(f_{11}), e(f_{12}), e(f_{13})\} + \sum\{e(f_{11}), e(f_{12}), e(f_{13})\} \\
 E(A_1) &= 25 + \max\{25, 25, 25\} + \sum\{25, 25, 25\} \\
 E(A_1) &= 25 + 25 + 75 = 125
 \end{aligned}$$

Whereas the evaluation of the node A_2 is:

$$E(A_2) = e(A_2) + \max\{e(f_{21})\} + \sum\{e(f_{21})\}$$

$$E(A_2) = 45 + \max\{30\} + \sum\{30\}$$

$$E(A_2) = 45 + 30 + 30 = 105$$

So, applying our method, we select the branch to the left, where the action A_1 is chosen to be optimized and performed.

3 Implementation

The Borregos team is based in the UvA Trilearn 2003 source code [4]. We have implemented some specific skills, like goal-shooting, considering some ideas presented in [10].

In the current prototype, our planning method is applied to the agent with possession of the ball; teammates just apply reactive heuristics aiming to help the player with the ball, like to stay far from opponents while attacking, etc. Of course, in principle the planning method could be applied to every single player, and most probably we will do it in future versions (see section 5).

To implement our strategy, we developed data structures for CPAs and instantiated data, then we implemented evaluation functions (both for benefit heuristic evaluation and for feasibility evaluation), action optimization, and parsing functions.

Our evaluation and feasibility functions make geometrical calculations and consider variables like traveled distance, opponent goal proximity, opponents density, teammates proximity, etc., that are reported elsewhere [12]. The feasibility function has been gradually adjusted in such a way that it corresponds to the average success rate of plays; this could be considered as a form of learning.

Another component links our decision mechanisms to a soccerserver command through the use of the UvA Trilearn code.

Figure 4 illustrates the way the planning method is implemented. First, according to the current situation of the game, the system checks the current possible actions that can be executed and make the instantiation, listing the CPA and the future actions of each member of the CPA. Next, current possible actions and its future actions are evaluated with respect to its feasibility and utility. After that, the best current possible action is optimized and finally executed. In the next cycle, when an agent has the ball, the process start again.

4 Experiments

We have been validating our strategy by playing games against our team disabling the use of future plays consideration. With the use of the proxyserver [6] we generated statistics to compare the performance of the team with the simultaneous tactics.

As a preliminary series of experiments, we ran several games between our team with the planning method ("Planning") and the same team without the planning method, and running a maximum-value method that chooses the action

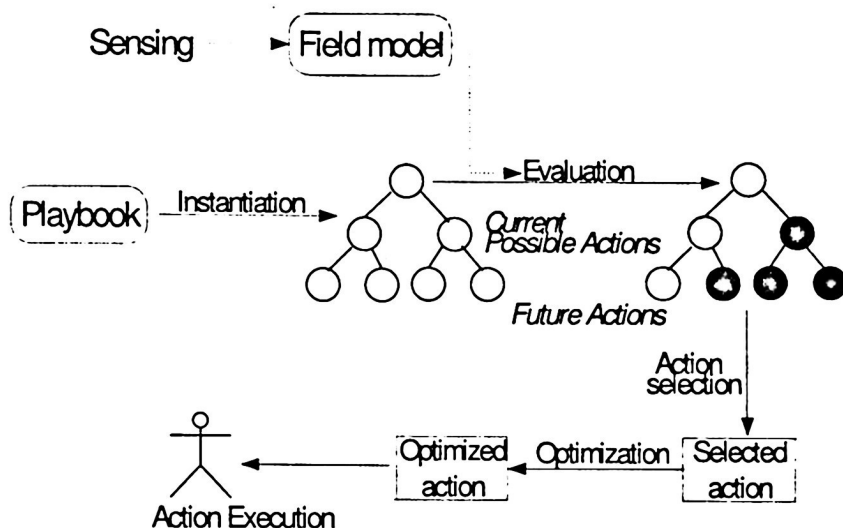


Fig. 4. Robocup prototype architecture

with the greater evaluation ("Nemesis"). Accumulated results are presented in table 1. The statistical parameters are the following:

- Total goals scored (Goal Scored)
- Average territorial advantage - ball in opposition half (territ)
- Average pass success rate (psucc)

As we can see in the table, the result was a slight, but clear advantage of the "Planning" team, both in terms of goals scored (26 scored, 22 received) and in terms of other statistical measures, like territorial dominance.

Table 1. Statistical parameters

Team	Goal Scored	territ	psucc
Planning	26	50.815	74.583
Nemesis	22	49.183	57.225

As we can see in Table 1, the goals scored by "Planning" team was clearly greater than the received ones (scored by "Nemesis"). Other parameters show a very slight advantage in territorial dominance, and also a great advantage in passing success rate (that is, number of passes that succeeded divided by the total number of passes). This last parameter can explain the global better behavior of the "Planning" team with respect to the reference "Nemesis" team.

We consider these as preliminary experiments, and currently we continue to refine the method implementation, in particular the evaluation functions, that are so critical to the planning method. More extensive experimentation is of course needed. But we consider an initial point is made about the possibilities of our planning method.

5 Conclusions and Future Work

We have presented a novel real-time planning method for highly dynamic domains with uncertainty, where possible actions evaluation is based not on the maximum in the search tree, but on a metric considering the quantity and quality of the options left available by the actions being considered.

An implementation of our method in the domain of the Robocup competition has been presented, which is at the heart of our "Borregos" team. Preliminary results, in the context of the SoccerServer Simulation league, are presented.

One critical aspect of our method is its high computational cost, because we need to perform many evaluations and then one optimization –involving even more evaluations. We relied on heuristics to reduce the search space for making the complexity manageable.

We have shown that our planning method has had promising results. We still have to work in the tuning of the evaluation, utility, and feasibility functions to increase the number of shoots to goal plays and the opportunities to score a goal.

Current research is focused on carefully evaluating our method's performance, for rigorous comparison against other planning methods, especially maximum-value based. Also, in the future, we want to apply machine learning techniques for parameters learning.

Acknowledgements. This work was supported by the Monterrey Tech's Research Grant CAT011.

References

1. James Allen, James Hendler, and Austin Tate. *Readings in Planning*. Representation and Reasoning Series. Morgan Kaufmann, San Mateo, California, 1990.
2. Michael Bowling, Brett Browning, and Manuela Veloso. Plays as effective multi-agent plans enabling opponent-adaptive play selection. In *Proceedings of International Conference on Automated Planning and Scheduling (ICAPS'04)*, 2004.
3. Mao Chen, Klaus Dorer, Ehsan Foroughi, Fredrik Heintz, ZhanXiang Huang, Spiros Kapetanakis, Kostas Kostiadis, Johan Kummeneje, Jan Murray, Itsuki Noda, Oliver Obst, Patrick Riley, Timo Steffens, Yi Wang, and Xiang Yin. Users manual: Robocup soccer server (for soccerserver version 7.07 and later).
4. R. de Boer and J. Kok. The incremental development of a synthetic multi-agent system: The uva trilearn, 2001.
5. Ahmad Farahany, Mostafa Rokooey, Mohammad Salehe, and Meisam Vosoughpour. Mersad 2004 team description, 2004.

6. Ian Frank, Kumiko Tanaka-Ishii, Katsuto Arai, , and Hitoshi Matsubara. The statistics proxy server. In Peter Stone, Tucker Balch, and Gerhard Kraetszchmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, pages 303–308. Springer Verlag, Berlin, 2001.
7. Lise Getoor, Greger Ottosson, Markus Fromherz, and Bjoern Carlson. Effective redundant constraints for online scheduling. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-97)*, pages 302–307, Providence, Rhode Island, July 1997. AAAI Press / MIT Press.
8. Frans Groen, Matthijs Spaan, and Nikos Vlassis. Robot soccer: Game or science.
9. Sven Koenig. Agent-centered search. *AI Magazine*, 22(4):109–131, 2002.
10. J. Kok, R. de Boer, and N. Vlassis. Towards an optimal scoring policy for simulated soccer agents, 2001.
11. J. Kok, M. Spaan, and N. Vlassis. Multi-robot decision making using coordination graphs.
12. Emmanuel Martinez. A real-time planning method for Robocup (in spanish). Master's thesis, Tecnologico de Monterrey, Mexico, 2005.
13. Patrick Riley and Manuela Veloso. Planning for distributed execution through use of probabilistic opponent models. In *IJCAI-2001 Workshop PRO-2: Planning under Uncertainty and Incomplete Information*, 2001.
14. Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 2003.
15. Peter Stone. Layered learning in multiagent systems. In *AAAI/IAAI*, page 819, 1997.
16. Peter Stone and Manuela Veloso. Using decision tree confidence factors for multi-agent control. In Katia P. Sycara and Michael Wooldridge, editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 86–91, New York, 9–13, 1998. ACM Press.
17. Peter Stone and Manuela M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.
18. M. Zweben and M. S. Fox. *Intelligent Scheduling*. Morgan Kaufmann, 1994.

Algoritmo de navegación de móvil robótico de trabajo en sociedad para mapeo de escenarios

Jesús Yaljá Montiel Pérez, Rodolfo Romero Herrera, Félix Serrano Talamantes

Escuela Superior de Cómputo, Instituto Politécnico Nacional
Unidad Profesional Adolfo López Mateos, Av. IPN S/N Col. Lindavista 07738
México, D. F.; becarios COFAA y EDD del IPN; yalja@ipn.mx

Resumen. Este artículo describe un algoritmo de navegación móvil aplicándolo en simulación robótica de trabajo en sociedad para lograr visitar en su totalidad un escenario delimitado.

1 Introducción

Actualmente la robótica ocupa un lugar importante en las actividades del hombre. Desde los procesos industriales hasta el hogar. Las barreras económicas y tecnológicas están siendo superadas y se está dando un auge en las técnicas y algoritmos de inteligencia artificial aplicadas a los robots. Los diseños mecánicos y algoritmos utilizados en los robots están siendo basados en la naturaleza [1]. Desplazamiento de un móvil basado en una araña, sistemas de estabilización neumático establecido a partir de la estructura de la cucaracha [2].

Por otro lado, algoritmos simples del comportamiento animal están siendo aplicados en la robótica y sistemas computacionales. El comportamiento grupal de las abejas exploradoras se está investigando en la aplicación de encontrar rutas óptimas para la transferencia de información en Internet y redes de comunicación [3]. Entonces, en una gran parte de la investigación en robótica móvil esta siendo orientada a la utilización y aplicación de características de seres vivos y este artículo describe el comportamiento de un algoritmo de navegación pseudo aleatorio en el cual existe una cooperación grupal por parte de los robots en la navegación en un escenario.

Un grupo de robots pueden trabajar cooperativamente para lograr un objetivo. Hasta el momento esta sociedad se basa en distribución del trabajo, especializándose cada robot en una tarea. Este tipo de organización ha demostrado ser eficaz, teniendo sus aspectos negativos, como el que si uno de los robots falla, el trabajo no puede ser finalizado.

2 Descripción del problema y algoritmo de navegación

Se tiene un escenario delimitado en el cual actúan tres móviles autónomos. Cada uno posee las mismas características como dimensiones físicas, velocidad y algoritmo de navegación.

El objetivo de los móviles es navegar a través del escenario y en conjunto visitar completamente a este escenario.

En la naturaleza se tienen a las hormigas y abejas, que tienen que salir al exterior de su hormiguero o colmena y buscar alimento. La navegación es hasta cierto punto aleatoria, por parte de las abejas buscan las flores que para ellas están marcadas por emisión ultravioleta de la luz solar. En el caso de las hormigas, buscan su alimento sin que se encuentre marcado por algún factor en especial.

Para ambos casos, se tienen varias hormigas o abejas que realizan este trabajo, es una colaboración para lograr un objetivo [4, 5].

Para nuestro caso, se utiliza una simulación de tres móviles los cuales tienen idénticas dimensiones y velocidad de navegación, así como el algoritmo de navegación. El algoritmo que se utiliza es relativamente simple y se puede observar en la Figura siguiente.

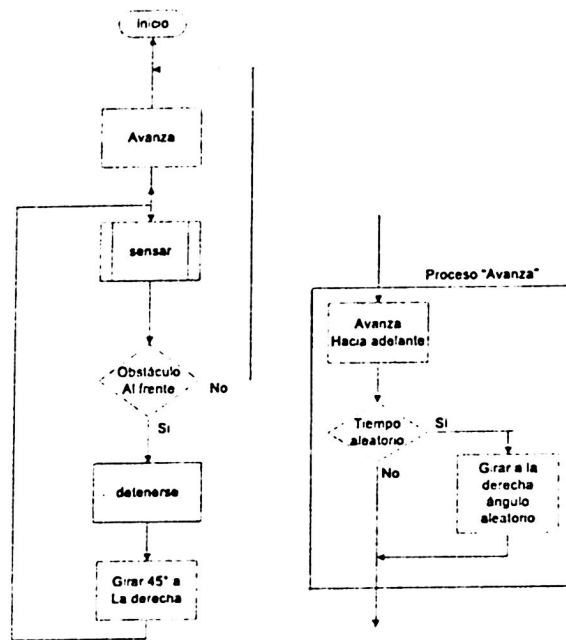


Figura 1. Escenario sin obstáculos donde el área de color negro pertenece al área visitada por los móviles.

Mientras el móvil no tenga obstáculos al frente, este avanza, si existe un obstáculo este se detiene y gira 45° a la derecha, si aún existe un obstáculo al frente, nuevamente vuelve a girar 45° a la derecha y así hasta tener campo libre al frente y poder avanzar. El algoritmo posee un tiempo aleatorio en el cual el móvil gira un ángulo aleato-

rio. Esto permite que no existan rutas cicladas y se quede el móvil recorriendo siempre el mismo camino.

3 Simulador

Se construye una simulación en lenguaje C, considerando 1, 2, 3 y 5 móviles dentro de un escenario sin obstáculos. Se aplica el algoritmo descrito en la anterior sección. Por facilidad en la programación se utilizan móviles circulares con diámetro de 20 píxeles. Para las pruebas realizadas con el simulador se estima una velocidad promedio de los móviles de 100 píxeles/seg. Al simulador se le plica que genere el tiempo aleatorio para realizar cambio de dirección del móvil en el intervalo de 1-100 segundos. También es posible almacenar en archivos de texto, información como % de área visitada por los móviles, tiempo, coordenadas de cambio de posición de los móviles, así como si velocidad.

Inicialmente se posicionan los móviles en posiciones aleatorias. El escenario sin visitar se marca con color rojo. Cuando un móvil se desplaza dentro del escenario, la zona por donde paso es pintada de color negro. Esta diferencia de colores se utiliza para obtener el porcentaje de área visitada. La Figura 2 muestra una prueba de simulación con 3 móviles después de algunos segundos de haber iniciado. Un aspecto en común para las pruebas realizadas es que, en los primeros segundos, se visita rápidamente gran parte del escenario, alrededor de 90%. Posteriormente los móviles pasan repetidamente por lugares ya visitados y no incrementa el porcentaje de área visitada.

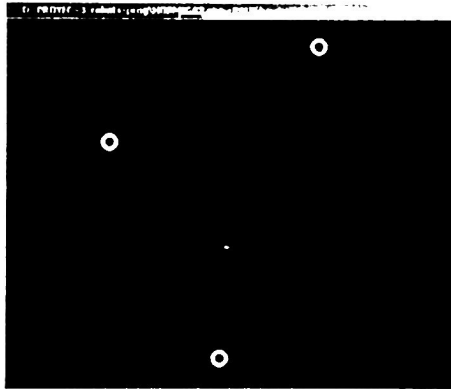


Figura 2. Escenario sin obstáculos donde el área de color negro pertenece al área visitada por los móviles.

Para visitar al menos el 98% de área del escenario, cada prueba tiene sus características. Mientras mayor sea el número de móviles más rápido se visita el escenario. La Figura 3 muestra este comportamiento. La línea para el comportamiento de un móvil, muestra un aumento de pendiente repentina en varias ocasiones a lo largo del

tiempo, estas variaciones de aumento de pendiente se deben al cambio de trayectoria en el tiempo aleatorio señalado en la simulación.

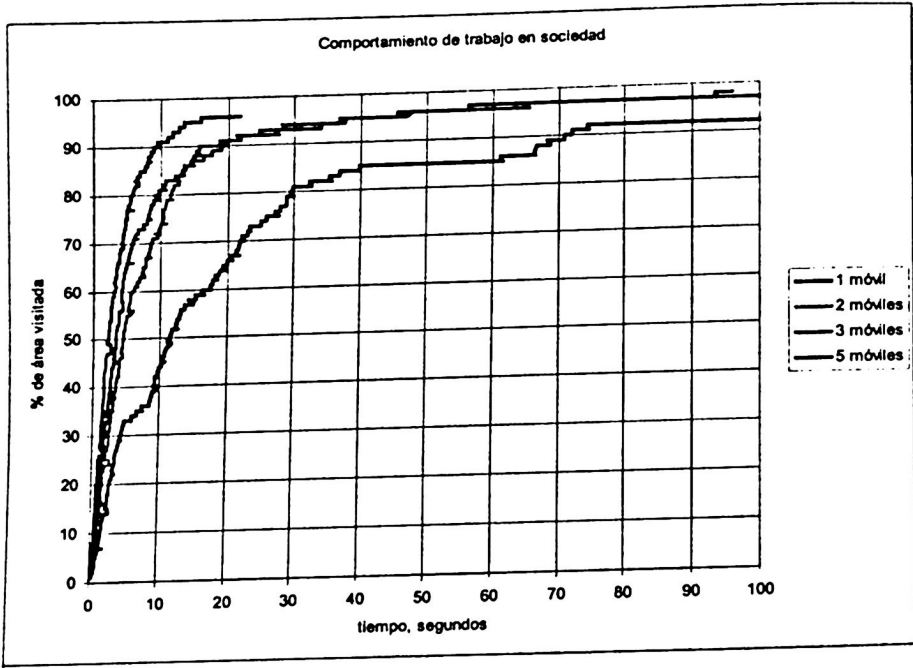


Figura 3. Comportamiento de %de área visitada con respecto al tiempo en función del número de móviles que intervengan.

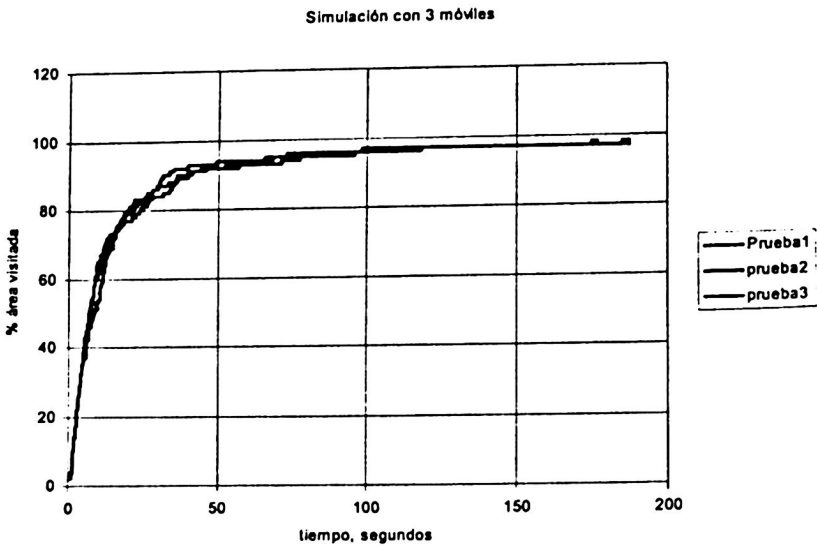


Figura 4. Similitud de simulación con 3 móviles.

El cambio de dirección aleatorio en un tiempo aleatorio, ayuda a no ciclar las trayectorias de los móviles.

Los tiempos de prueba se acortan en función al número de móviles utilizados en la simulación. Para el caso de la simulación con un móvil, el proceso de simulación se ejecuto aproximadamente por 90 minutos, sin superar el 97% de área visitada.

Cabe señalar que para cada situación de número de móviles se realizaron múltiples ejecuciones del software, comportándose de manera similar y teniendo sus cambios de pendiente en diferente tiempo, Figura 4.

4 Modelo probabilístico

Un modelo probabilístico de navegación utilizando las reglas antes descritas se describe a continuación y será válido para un móvil.

Suponiendo un móvil de área rectangular A_0 , recorriendo un escenario, podríamos conocer el área que ha recorrido el móvil si se conoce el ancho de este y la distancia que ha recorrido en un tiempo dado. Suponiendo que el móvil se desplaza con movimiento lineal rectilíneo, se tiene:

$$A = xd \quad (1)$$

Siendo x la distancia que recorre el móvil y d el ancho del mismo.

Ahora bien, si se tiene un objeto en el escenario de manera tal que al chocar el móvil con este, tuviera una desviación de 45 grados a la derecha con respecto a su dirección de avance, el área que recorrería el móvil se calcula como:

$$A_T = A_1 + x_2 d - A_0 \quad (2)$$

Donde A_1 es el área recorrida por el móvil antes de colisionar con un obstáculo, x_2 es la distancia recorrida después de la colisión. El área del móvil A_0 es necesario restarla ya que al girar el móvil debido al obstáculo y empezar nuevamente el recorrido se volvería a sumar esta área por la intersección de de las dos trayectorias, antes y después de la colisión.

Generalizando, si se detectan m obstáculos dentro del escenario, se tendrán $m+1$ recorridos del móvil dentro de la superficie. En principio el área total recorrida será:

$$A_T = \sum_{i=1}^{m+1} A_i - mA_0 \quad (3)$$

en esta aproximación aún no se han considerado los cruces de trayectoria. Si se tienen n de estos cruces se tendrán que restar del área total considerada, entonces la expresión 3 queda:

$$A_T = \sum_{i=1}^{m+1} A_i - (m+n)A_0 \quad (4)$$

Las paredes del escenario se consideran como un obstáculo. Por otro lado ya que se ha considerado un movimiento rectilíneo uniforme, es posible usar la ecuación que describe tal movimiento para obtener el área visitada en función de parámetros conocidos, es decir la velocidad del móvil. Y dejar la expresión en términos del tiempo de recorrido.

La expresión de movimiento rectilíneo uniforme es:

$$x = vt \quad (5)$$

Donde v es la velocidad del móvil y t el tiempo de recorrido. Entonces, la expresión (4) se transforma en:

$$A_T = \sum_{i=1}^{m+1} vt_i d - (m+n)A_0 \quad (6)$$

la velocidad es una constante y definiendo al tiempo como $T = \sum_{i=1}^{m+1} t_i$ la ecuación anterior se convierte en

$$A_T = vTd - (m+n)A_0 \quad (7)$$

De donde se puede obtener el tiempo total de recorrido

$$T = \frac{A_T + (m+n)A_0}{vd} \quad (8)$$

Por otro lado, en el algoritmo existe un factor de tiempo aleatorio que modifica la trayectoria del móvil. Este término puede ser representado mediante la teoría de caminata al azar. Entonces para un objeto que se mueve k pasos en una dirección, la posición después de esos pasos será [6]:

$$\begin{aligned} x &= (k - (N - k))s \\ x &= (2k - N)s \end{aligned} \quad (9)$$

donde S es el tamaño del paso y $N-k$ el número de pasos hacia el frente. Para nuestro problema el tamaño de paso se puede considerar como la longitud total del móvil y cada vez que avanza el móvil se da un paso. Considerando que el móvil es cuadrado y tiene un área de A_0 , se puede suponer que $s = \sqrt{A_0}$ y $N=n$ el número de ocasiones en que se repite un área. De tal manera que

$$x = (2k - n)\sqrt{A_0} \quad (10)$$

por otro lado, en la caminata al azar se puede considerar que está o no en la posición x determinada por una probabilidad de tipo binomial, así que ésta estará dada en términos antes considerados:

$$P = \binom{n}{k} p^k q^n = \frac{n!}{(n-k)!k!} p^k q^n \quad (11)$$

con p la probabilidad de dar k pasos y $q=1-p$. k se encuentra de despejarle de (10):

$$k = \frac{n}{2} + \frac{x}{2\sqrt{A_0}} \quad (12)$$

Además de

$$n-k = \frac{n}{2} - \frac{x}{2\sqrt{A_0}} \quad (13)$$

Esta expresión puede dejarse en términos de la expresión (8), así que la probabilidad se puede obtener en función del área recorrida y la misma expresión podría quedar en términos del tiempo de recorrido del móvil.

$$P = \frac{n!}{\left(\frac{n}{2} - \frac{x}{2\sqrt{A_0}}\right)! \left(\frac{n}{2} + \frac{x}{2\sqrt{A_0}}\right)!} p^{\frac{n}{2} + \frac{x}{2\sqrt{A_0}}} q^n \quad (14)$$

Esta expresión es la probabilidad de que el móvil se encuentre en la posición x y se recorra un área en un tiempo dado.

Experimentalmente en base a las simulaciones realizadas se encuentra mediante regresión polinomial y análisis de trayectorias, Figura 4, que el número de obstáculos está expresado mediante:

$$m = 0.43T^2 + 0.4T + 0.08 \quad (15)$$

Mientras que el número de cruces del móvil por áreas visitadas con anterioridad con respecto al tiempo se expresa con:

$$n = 0.1T^4 + 1.25T^3 - 3.93T^2 + 4.05T + 0.0002 \quad (16)$$

Además de que se tiene que el ancho del móvil, $d=20$ pixeles, y un área $A_0=400\text{pixeles}^2$ y una velocidad promedio de $v=100$ pixeles/seg.

Del análisis de datos de trayectoria, con respecto a la probabilidad de que el móvil de k pasos o secciones de trayectoria, tienen la misma probabilidad, 40% aproximadamente, de que ocurra tanto para cortas longitudes (30 pixeles) como para largas (180 pixeles); teniéndose baja probabilidad para longitudes de trayectorias que se encuentren entre estas longitudes.

La siguiente tabla muestra los resultados de la evaluación del modelo teórico con respecto a los datos obtenidos de la simulación.

Tabla 1. Resultados de la Evaluación del Modelo Teórico.

tiempo	% área cal.	% área sim.
1	2.26639804	1
2	4.91038983	9
3	7.55438162	17
4	10.1983734	25
5	12.8423652	31
10	26.0623242	54
20	52.5022421	69

5 Mapeo de escenarios

De manera experimental se utilizan 3 móviles autónomos con el algoritmo de navegación programado en un PIC. El móvil posee los sensores necesarios para detectar obstáculos y con el algoritmo se deciden trayectorias de navegación. Cada móvil tiene comunicación unidireccional con una computadora mediante un modulo de radio frecuencia. Con lo cual, se envían datos de avance, trayectoria ejecutada y detección de obstáculos por cada móvil a una computadora. Esta se encarga de analizar y sincronizar datos de tal manera que se puedan registrar puntos de detección de objetos y graficar estos para generar a lo largo del tiempo un mapa del escenario donde se desenvuelven los móviles.

La computadora recibe los datos de cada móvil. Con estos se crea un mapa indivi-

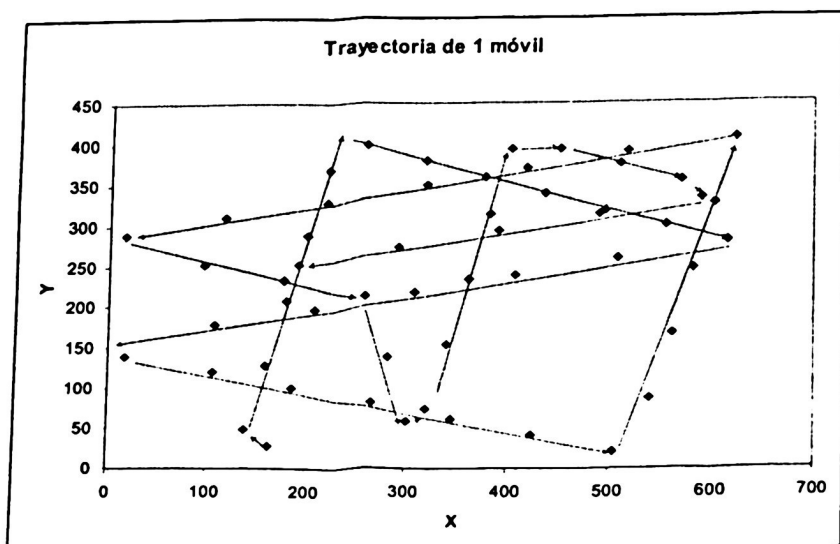


Figura 5. Trayectoria de un móvil ejecutada en un tiempo de 5.19 segundos.

dual de cada móvil. Mediante una red neuronal con entrenamiento hacia atrás, back-propagation, se analizan los mapas individuales, buscando coincidencias, de tal manera que una vez que se tenga los suficientes datos para poder encontrar coincidencias entre mapas con un error. Los gráficos que representan estos mapas se mezclan dando un mapa resultante.

Se utiliza un entrenamiento de la red neuronal con 10 posibilidades de coincidencia, tomando como referencia, esquinas y dimensiones de objetos. El tipo de red neuronal utilizado permite tener salidas de coincidencia aun con errores grandes. Lo cual da la posibilidad de que en tiempo de ejecución del sistema, se pueda corregir el mapa generado al inicio de la obtención de datos.

Por las características de diseño de los móviles, el trazar un mapa de un área de $5m^2$ tiene una duración aproximada de 15 minutos, Figura 5.

6 Conclusiones

Mediante la simulación en un escenario con 1 o más robots, se muestra que el algoritmo propuesto, aunque simple logrará que el móvil visite toda el área donde se encuentre. El factor a considerar es la rapidez con que lo realice, ya que varios móviles cubrirán el área del escenario de una manera más rápida que siendo menor su número.

El algoritmo implica que el móvil repetirá trayectorias en el escenario, pero el móvil no tiene que recordar por donde ya ha pasado; y este aspecto simplifica cuestiones de memoria y algoritmos complejos de navegación, lo cual repercute directamente en el costo de un móvil real, con elementos como microprocesadores, memoria, tiempo de respuesta a nuevos datos, etc.

Otra característica importante de este algoritmo es que, el móvil no necesita conocer previamente el escenario.

Este simple algoritmo, ejemplifica el comportamiento de las hormigas, las cuales simplemente por olfato y tacto navegan en su ambiente. Esto implica que los móviles a construir con este tipo de algoritmos tendrán entrada de datos, sensores de presencia, como micro switch o sensores ópticos de corto alcance.

El modelo teórico únicamente introduce aspectos superficiales de la teoría de caminata al azar, lográndose una aproximación de estimación de área en función de tiempo esencialmente de tipo lineal. Mientras que las simulaciones muestran un comportamiento más bien logarítmico. Y a esto se deben los errores mostrados en la comparación de datos de simulación y estimados. Por consiguiente, el siguiente paso con respecto a un modelo teórico es la introducción de un mayor número de conceptos de caminata al azar.

Este tipo de algoritmo se utiliza en la actualidad para barredores robot de piso caseros. Donde no importa tanto el tiempo en que realice el trabajo, sino que cubra la totalidad de superficie del escenario, sin que previamente conozca éste.

Por otro lado este algoritmo se encuentra en fase de investigación para la creación de mapas de un escenario. Donde se aplican las características de que el móvil no conoce el escenario, el tiempo no es factor predominante.

En esta aplicación el móvil posee el algoritmo descrito en este artículo y el sensor de obstáculos registra las coordenadas de este y en otra ventana se grafican los puntos donde se observó tal obstáculo.

Para finalizar, este algoritmo, aún que no nuevo, puede tener una gran variedad de aplicaciones, simplificando requerimientos de hardware en los móviles y simplificando la programación de estos.

References

1. Randall D. Beer, Roger D. Quinn, Hillel J. Chiel, and Roy E. Ritzmann; Biologically Inspired approaches to robotics"; Communications of the ACM; Vol. 40; No.3; March (1997); 31-38.
2. E. V. Krishnamurthy; V. K. Murthy, Vikram Krishnamurthy; Biologically Inspired Rule-Based Multiset Programming Paradigm for Soft-Computing; CF'2004 ACM; april (2004); 140-149.
3. Bonabeau, Théraulaz; "Swarm Smarts"; Scientific American; March (2000).
4. Tucker Balch, Zia Khan and Manuela Veloso; Automatically Tracking and Analyzing the Behavior of Live Insect Colonies; Agents 01 ACM; 28 May (2001); 521-528.
5. Lam, Y.K. Wong, E.K. Loo, C.K.; Explicit communication in designing efficient cooperative mobile robotic system; Proceedings IEEE International Conference on Robotics and Automation, 2003; Vol. 3; 14-19 Sept. (2003); 3869 – 3874.
7. A. M. Odizko; Search for the maximum of a random walk; ACM; (1994); 336-345.

New Motion Prediction Algorithm Invariant to Rotation and Occlusion

José Tomás Tello¹, Oleg Starostenko¹, Gennadiy N. Burlak²

¹ Computer Science Department, University de las Américas,
Cholula, Puebla, 72820, Mexico
oldwall@mail.udlap.mx

² Center for Research on Engineering and Applied Sciences,
Autonomous State University of Morelos, Cuernavaca, Mor. Mexico
gburlak@uaem.mx

Abstract. The novel algorithm for motion estimation and prediction of objects in dynamic scene by processing a limited sequence of images is presented in this paper. The proposed hybrid algorithm is based on computing differences between consecutive frames of video stream for fast detection of objects in motion by processing of their time-varying edges. For quantitative prediction of motion and computing of its characteristics, the interpolation technique has been used in combination with the motion stabilization technique useful for analysis of irregular displacement of objects. The invariance to occlusion and rotation of object in motion is achieved by proposed double envelope approach. The principal goal of proposed algorithm is the development of simple and efficient facilities for motion prediction when the possible routes may be computed even the objects in motion has occlusions and rotation. The introduced algorithm has been tested and evaluated.

1 Introduction

An artificial vision system for motion detection and prediction consists of three basic subsystems: the preprocessing of digital images (segmentation and description), the patterns (objects) recognition and persecution, and prediction (computing the next possible state of the traced object within the scene). A key module for analysis of translation of rigid objects in sequences of images is pursuance unit which principal problem is the quantity of the processed data due to necessity of bit by bit image inspection for detecting the changes in the position of objects. The input of a system for analysis of dynamic scene is a frame sequence where each frame represents the scene at time t . It assumes that image is obtained using camera located at the origin of a three-dimensional coordinate system. The projection used in this observer-centered system may be either perspective or orthogonal. It is also assumed that since the frames are usually taken at regular intervals, t represents the t -th frame of the sequence, rather than the frame taken at absolute time t . Any perceptible by eyes motion in a scene consists of some changes in the frame sequence within video stream. Motion characteristics can be analyzed if such changes are detected. A good quantitative

estimation of motion components of an object may be obtained if it is restricted to a parallel motion to the image plane.

The sequence of images contains necessary information about dynamic scene and usually it is defined by optical flow or motion field, which can be estimated by well-known methods [1], [2]. The obtained information about motion can be used as input of different subsequent processes including motion detection, motion compensation, motion-based data compression, 3-D scene reconstruction, autonomous navigation, analysis of dynamic processes in scientific applications [3], [4]. Normally the optic flow and the motion field (object displacement vector) are different, but they are based on the similar motion characteristics and with certain approximation their quantitative properties can be equal [5]. Thus, the optic flow equation is used for computing the motion field, and the motion characteristics are used for optic flow construction. Usually, the computing the motion characteristics can be obtained more quickly on base of processing the object edges or principal corners instead of analysis of intensity variations or complete object correlation in consecutive frames [6].

Sometimes the quantitative estimation of motion characteristics is provided by block correspondence techniques where estimation of the best similarity of features or regions within consecutive frames at high level (configuration, regions analysis) or low level (corners, patterns, color changing analysis) is obtained. In these approaches the coordinates (x', y') of the center of analyzed pattern are found after pattern translation from coordinates (x, y) during the time interval Δt [7]. Thus, the same pattern is used within consecutive images as reference one. It allows overcoming the problem of progressive increment of compared patterns but aggregates accumulative error proportional to time function. This process can be modeled by the brightness function of analyzed pattern calculated according the equation:

$$C(x, y, x', y') = \sum_{-M/2}^{M/2} \sum_{-L/2}^{L/2} \{E(x+m, y+l, t-\Delta t) - E(x'+m, y'+l, t-\Delta t)\} \quad (1)$$

where $E(x, y, t)$ is the brightness value of the pattern at the time t , M and L are two dimensions of the pattern toward the axis x and y respectively. Computing the motion characteristics is provided by estimation of the displacement vector F (motion field vector) during comparison of the similar $C(x, y, x', y')$ functions for analyzed and reference patterns as it shown in Fig. 1.

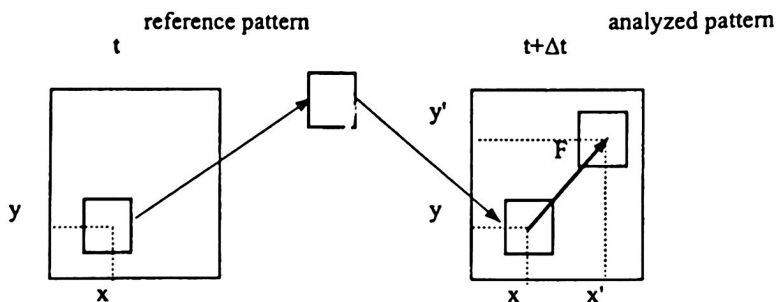


Fig. 1. Blocks correspondence technique for dynamic scene analysis.

On the base of absolute value of F and time interval Δt between two consecutive images the speed of object translation is calculated. The disadvantage of this technique is dependence on the dimensions of pattern in motion [8].

To predict the motion of dynamic objects the additional techniques must be used. These techniques are based on computing the optic flow or motion fields where the tracing and prediction of objects in the consecutive frames may be described by mathematical equations. The powerful technique is interpolating approach which provides high efficient and fast description of well-defined trajectories. In case of irregular motion frequently occurred in dynamic scenes the motion compensation or stabilization technique must be applied. It is important to mention that proposed motion prediction approach must operate efficiently when occlusion or rotation of objects in motion is appeared.

2 Motion Prediction Techniques

There are a lot of reports about well-known methods for border estimation which can be classified as it follows: approaches based of computing the probability of intensity function of objects in motion [9], gradient-based methods [1], block correspondence techniques [7], spatial-temporal volumetric filtering methods of the 3D images [4], phase-based approach applying the speed sintonization filters [3], USAN circular mask area manipulation technique [10], neural networks based approach [11]. But the border detection techniques require a long process of pixel by pixel analysis for each frame and then recognition of the detected borders, forms, etc.

The fastest and most obvious method for detecting changes between two frames consists in comparison the corresponding pixels of two frames to determine whether they are similar. In this approach, a binary difference picture $DPjk(x, y)$ between frames $F(x, y, j)$ and $F(x, y, k)$ is obtained by applying the following equation

$$DPjk(x, y) = \begin{cases} 1, & \text{if } |F(x, y, j) - F(x, y, k)| > \tau \\ 0, & \text{in other cases} \end{cases} \quad (2)$$

where τ is a threshold. The difference and accumulative difference presents the well-interpreted regions with changes usually due to motion of objects. This approach is very sensitive and permits to detect very small displacement of object. The disadvantage is dependence on correct selection of threshold τ .

There are some efficient motion prediction approaches which may be mentioned as

a) In the interpolation approach the motion predictor tracks the screen space location of points on an object. Each object in the scene has associated transformation matrix, which transforms object from model space to screen space in frame [4]. During the initial rendering of the image the matrix transformation number of the object visible in each pixel is stored - each object has a unique transformation number. If more than one object is visible in the pixel, for example in the case of transparent objects, then it would be possible to store multiple transformation numbers with the

relative contributions of each object to the total pixel intensity. However, the actual implementation of the algorithm stores only the transformation number of the front most object visible in the pixel.

b) The Brownian motion model is applied in the statistical approach where the Bayesian filters for the tracking of object with easy prediction of motion are useful. The Brownian model is extremely conservative and does not attempt to model the dynamics of irregular motion [12]. This is often a poor estimate of the actual distribution over a possible position because objects with easy predicted displacement do not move randomly.

c) The Kalman's filter is the standard technique using prediction to improve state estimation over the time. They have been successfully used to track, for example, the boundaries of hands in image sequences. However, these filters need preprocessing, or data association, to determine which measurements in the image should be used to update the motion model. The filter is distinguished for its ability to predict the state of a model in the past, present and future, still when the precise nature of the modeling system is ignored.

d) Multi-hypothesis motion compensated prediction predicts a block from a weighted sum of multiple reference blocks in the frame buffer. By efficiently combining these reference blocks, it can provide less prediction errors so as to reduce the coding bit rate [13]. Usually long-term memory motion compensation generates only one motion vector for each macro-blocks, so this approach allows more than one motion vector.

After analysis of efficient well-known methods the hybrid approach for motion prediction is introduced, particularly the proposed algorithm is developed taking into account advantages of the interpolation technique, block correspondence approach for tracing of objects in motion invariant to occlusion and rotation, and motion compensation method for irregular motion stabilization.

3 Proposed Algorithm for Motion Prediction

The classical but enough efficient approach of motion prediction is based on using the interpolation technique when expected values are computed by applying the function derived on base of known values. Conceptually, the interpolation process has two stages: fit an interpolating function to known data and evaluate the errors of prediction using interpolating function at new target. The proposed algorithm for fast motion prediction based on application of subtraction operator and Lagrange polynomial interpolation may be described as it follows:

a) The input images of normalized size are the frames of video streams to be converted to gray scale images.

b) The detection of objects in motion using limited sequence of images is obtained by analysis of changes $DPjk(x,y)$ between consecutive images $F(x,y,j)$ and $F(x,y,k)$ according the equation (2) where threshold τ is selected with respect to necessary quantity of segments representing the object in motion.

c) The objects in motion detected by subtraction operator on step b) are wrapped up by circular envelope. It is used as description of the same object in the following frame without taking into account the small variations of detected objects in motion. Using the envelope instead of complete borders of object reduces the errors of subtraction operator and simplifies the manipulation with objects with occlusions.

d) The center of circle (envelope) for object in motion is computed. The circle center represents the gravity center of the object and it is used as coordinates of object within the image. If there is more than one object in motion, each one is represented by center of its envelope and the general overall envelope for all these object is defined. The overall envelope is used on step g) if the occlusions of the objects with other ones in motion are detected. The occlusion is automatically detected when the size of overall envelope is less than the double size of envelopes representing the objects in motion.

e) The generation of interpolating polynomial is provided on base of position of envelopes corresponding to the same object in video stream. The interpolating polynomial of degree $N-1$ for the N points (envelope coordinates) $y_1 = f(x_1)$, $y_2 = f(x_2)$, ..., $y_N = f(x_N)$ is given by Lagrange classical equation:

$$P(x) = y_1 L_{n,1}(x) + y_n L_{n,n}(x) = \sum_{k=0}^n y_k L_{n,k}(x) \quad (3)$$

f) The extrapolated predicted object position $P(x+1)$ is computed on base of results of previous step. In this way the accuracy of predicted and real translation is evaluated computing their absolute and relative errors.

g) In case of occlusions between envelopes the additional analysis of the object in motion is applied. This analysis is based on detection of the object borders and their continuous tracking using the Sobel spatial gradient operator [14]. Obtained borders in motion are traced for detection of the displacement vector according block correspondence approach as it shown in Fig.1 and the Segment and Neighbors Matching method proposed by authors in [15]. This method is based on the concept of fuzzy sets and operates with membership grade as principal criteria for including the analyzed element to the set of fuzzy segments which compose the pattern. This step permits to reduce the number of similarly oriented segments, increment the speed of tracking the objects in motion, and predict their next position in case of occlusion with other ones. For evaluation of the proposed algorithm some applications have been designed and tested.

4 Results and Discussion

According to proposed algorithm the objects which manifest motion are detected by clustering the areas tracked them in the sequence of frames in video stream. Some experiments for estimation of efficiency of proposed algorithm have been done taking into account that the video stream is captured by a single fixed camera and computing the characteristics of relative motion between objects is not provided. The Fig. 2a) shows the displacement of two objects and the detecting them by computing the dif-

ference of two superimposed frames. Also an envelopes of two objects and overall envelope as the result of clustering the area with motion are shown.

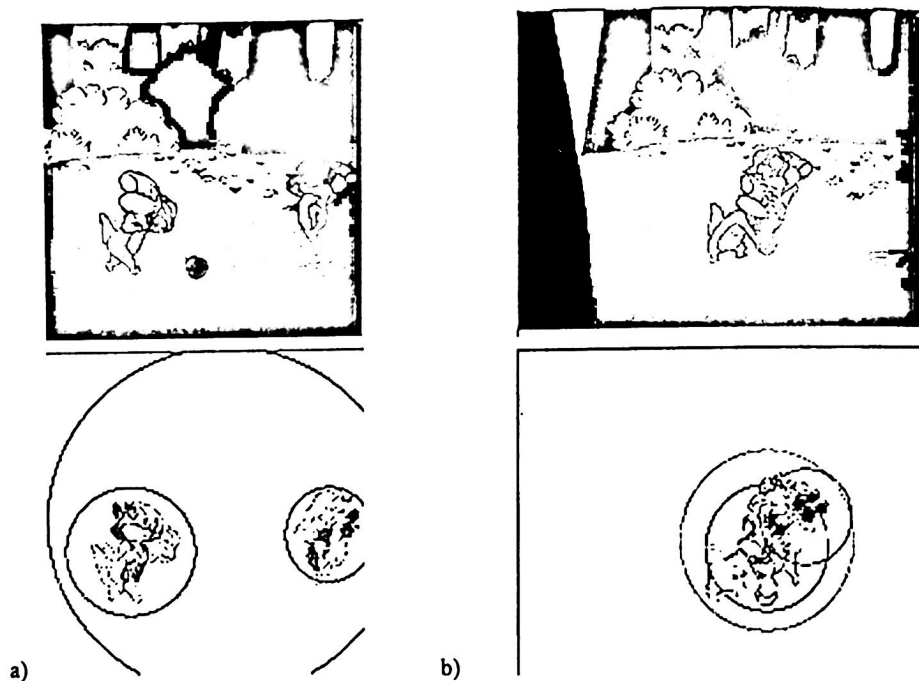


Fig. 2. Detection of object in motion a) the objects in motion without occlusion are presented by separate envelope. b) the objects in motion with occlusion are detected by intersection of envelopes. The overall envelope shows the area with objects in motion.

A small circle in Fig.2 represents the envelope and rounds the area where there is an object in motion. Their center is defined by the coordinates computed as average value of coordinates of the segments in motion. In Fig. 2b) the overall envelope has dimensions about the size of envelopes of single objects. In this case the step g) of the proposed algorithm is applied.

If the periodical motion is detected the prediction becomes effective when at least one period of oscillation is described by interpolation function. The irregular motion can not be predicted with high accuracy because it is impossible to generate exact interpolation function. But anywhere, the greater number of processed frames permits to reduce the number of predicted positions of object in next step. In this case it is possible to talk about so-called motion stabilization function. The stabilization heuristic function is obtained as a set of possible routes of the object in motion when the error between the predicted and described by interpolating function displacement vector goes to zero. Taking into account interpolating function the probability of each route is computed and more probable one is selected. This algorithm implements the dynamic prediction because the next object position depends on particular instant of

time and the result of application of interpolation and stabilization functions at this moment.

In the Fig. 3 the stabilization of the interpolation process is shown. The white line presents the direction of object displacement described by interpolating polynomial, the black one shows the direction of predicted motion computed on base of analysis of real position of the object in previous frames. The white circle defines the position of the object in predicted step.

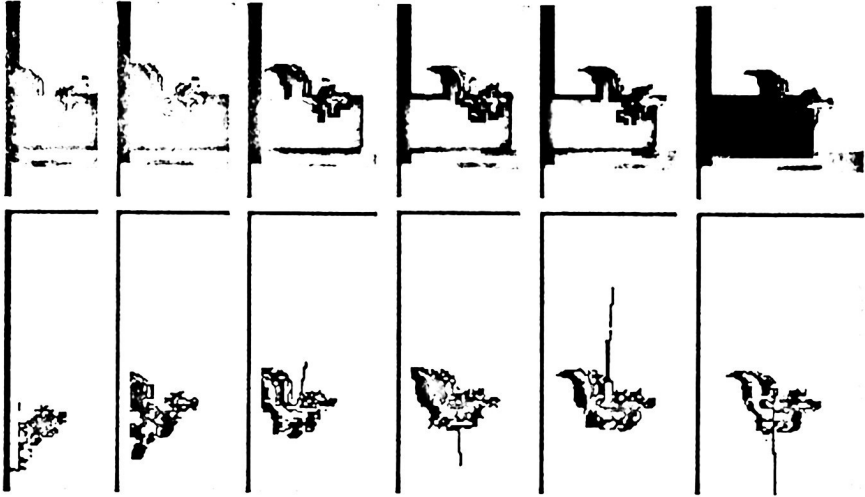


Fig. 3. Oscillation of the interpolating polynomial until reaching the stabilization.

The stabilization is obtained when the predicted and computed by interpolating polynomial displacement vectors have the similar direction with the certain error, for example, the displacement is less than 45° . The last frame shows the fact that predicted and computed displacement vectors have the similar direction. The error between them is reduced by processing more frames taken for analysis and more terms in interpolating polynomial. That minimizes the number of possible routes on the next step.

In the dynamic scenes with quasi-linear object displacement as it shown in Fig. 4, the motion description is predictable with smaller number of terms in interpolating polynomial and motion stabilization is obtained after first two frames.

The development of proposed hybrid algorithm for motion prediction has the principal objective to apply the advantages of some high-performance methods. From the experimental data it is clear that the interpolating approach provides high efficient and fast description of well-described trajectories. The motion compensation or stabilization technique is applied in case of irregular motion frequently occurred in dynamic scenes that reduce the error of the predicted route. It has been proved that high accuracy object tracing is provided fast enough by using the block correspondence

method that permits the prediction of motion in case of occlusion of objects and the motion invariant to their rotation. This is the principal contribution of our report.

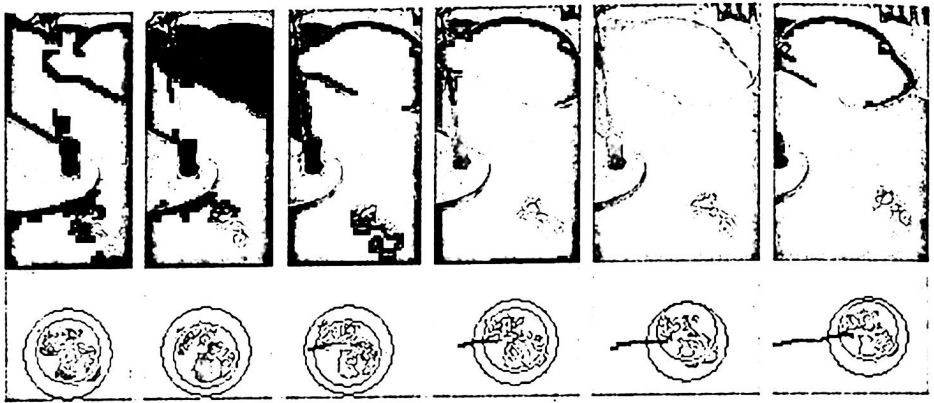


Fig. 4. Description of quasi-linear motion. The real and predicted motion is presented by black and white lines. The black line on the left part of processed image shows real accumulative translation from the initial point. The white line shows the real translation with respect to previous frame, and the black line on the right of the white line presents the direction and the distance of the predicted translation.

In order to evaluate the efficiency of the proposed approach the analysis of predicted by Lagrange extrapolation and real translations of object are compared as it shown in table 1.

Table 1 Analysis of errors for real and predicted motion. The row in bold letter with real coordinates $x=100$, $y=128$ shows the lowest error due to the lowest speed of object in motion.

x real	y rea	x pred	y pred	Δx	Δy	δy
20	82	20	61	0	21	0.3442
40	108	40	92	0	16	0.1739
60	125	60	113	0	12	0.1061
80	132	80	125	0	7	0.056
100	128	100	127	0	1	0.0078
120	116	120	119	0	3	0.0252
140	93	140	101	0	8	0.0792

The real x and y coordinates of object within each frame are presented in columns x_{real} and y_{real} . The relative predicted positions of object are computed and then adjusted to absolute coordinates as it shown in columns x_{pred} and y_{pred} . These values now may be compared, therefore the absolute additive errors $\Delta x = |x_{\text{real}} -$

x_{pred} , $\Delta y = |y_{real} - y_{pred}|$ (column Δx and Δy) and relative error $\delta y = (y_{real} - y_{pred}) / y_{real}$ (column δy) are computed. The image corresponding to motion characteristics presented in table 1 is shown in Fig. 5 where the parabolic motion of the shot ball with initial acceleration has been analyzed.

It is important to mention that the absolute and relative errors of object's translation along x axis are equal to zero because the x component of motion is uniform due to constant speed. But the y component of motion has acceleration and the predicting model produces the errors of value proportional to magnitude of the acceleration. This problem may be solved by more frequent sampling of frames during intervals with acceleration of high magnitude. In the Fig. 5 the positions of object corresponding to its real positions in two consecutive frames are shown on one processed frame as two white balls. The overall black circle shows the area where there are objects on motion in two consecutive frames and small one shows the possible position of object in next the frame. It may be mentioned that the algorithm provide exact prediction of position (the direction of the real and predicted displacement practically are matched) due to the route is well-defined function.

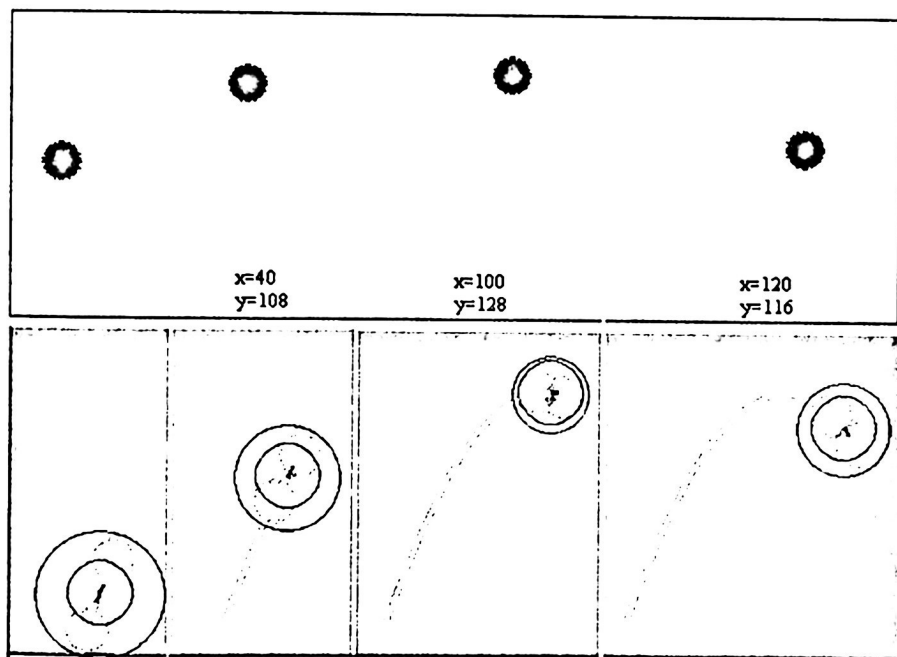


Fig. 5. Example of the parabolic motion of object and its some predicted and real position which have been presented in the Table 1. The thin white line on the left of the object shows accumulative real displacement. The thick white line presents the direction of real translation with respect to previous frame and the black thick line is direction of predicted displacement.

5 Conclusions

This paper presents the early research results in development of motion analysis and prediction facilities. There are a lot of techniques for motion prediction but they are not so efficient and fast enough for real-time applications which require the great quantity of data preprocessing of video streams. The proposed algorithm is one of possible solution for estimating and predicting the motion of objects within dynamic scenes based on interpolation of their routes.

For reduction of processed data the small quantity of the segments represented the object in motion are processed by tracking them using double envelopes useful for route construction and prediction. For exact prediction of irregular motion the motion stabilization for interpolated displacement vectors are proposed. The advantage of manipulation with motion stabilization function permits to apply same approach to irregular motion as it is provided for periodical of well-described translation. The disadvantage of approach for motion stabilization is a long time period of computing the new possible routes until the difference in direction between predicted and described by interpolating function displacement vectors reaches the minimum value.

Moreover, the proposed algorithm solves the problem of object tracking and their motion prediction when the occlusion between objects is presented. The processing objects borders is applied using the Block correspondence and Segment and Neighbors Matching method which permit exactly to trace the segments on base of fuzzy sets. It takes additional time but this method will be used only during the occlusions. The motion without occlusion is processed by analysis of envelope translation proposed for representation of object without taking into account small variations of it form or color. The use of envelopes provides simple and fast detection and tracing of object in motion with rotation. The principal restriction of proposed approach is that the computing the characteristics of relative motion of objects is not provided.

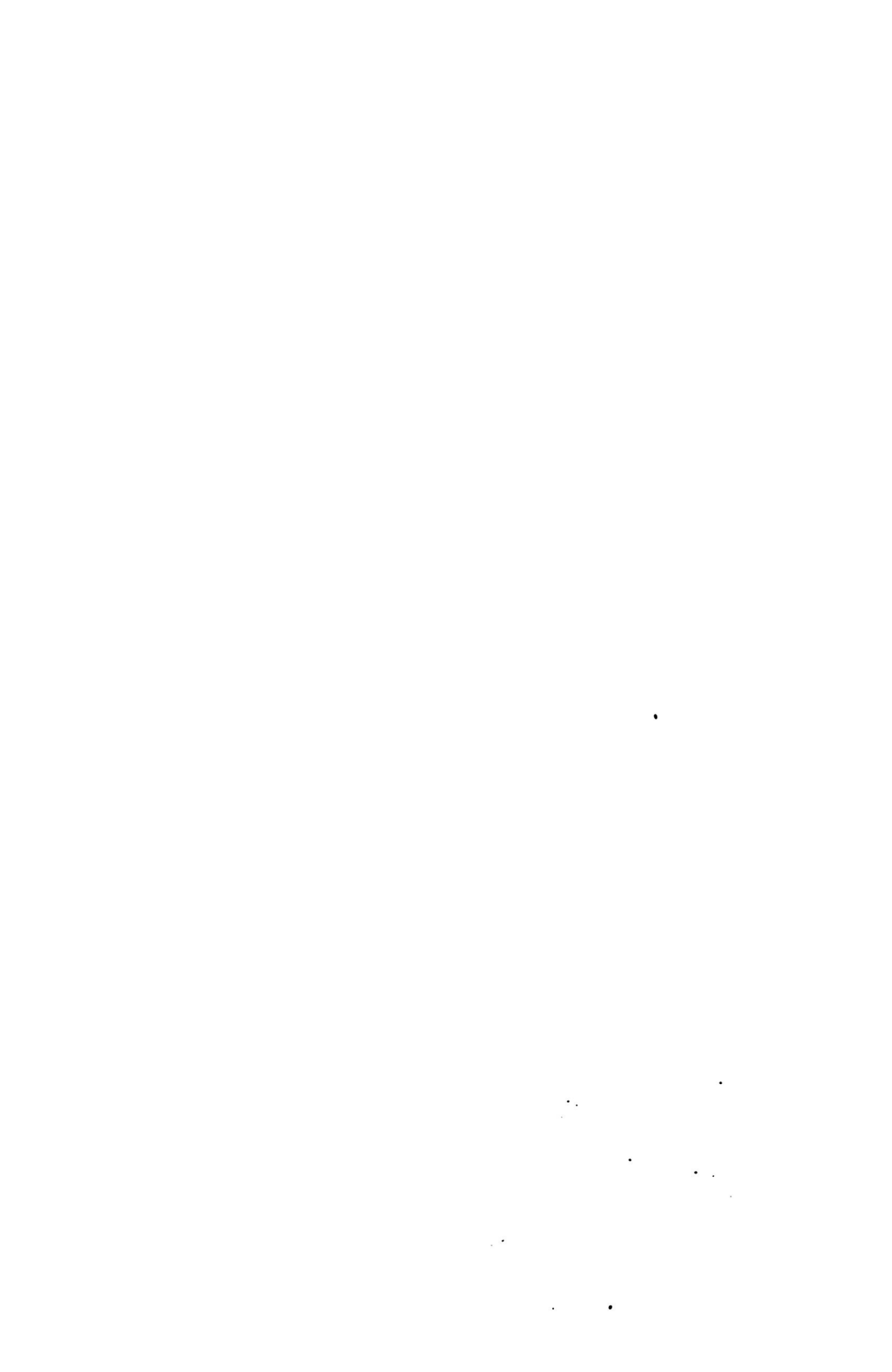
Preliminary experiments with proposed algorithm show that the faster motion prediction is achieved due to the lower quantity of processed data taking into account irregular motion, possible objects rotation and occlusion.

In the future works a generalization of the interpolation procedure will be made. It is important task to evaluate and compare the obtained results with some efficient approaches such as the Kalman's filter and the probabilistic correspondence of blocks. The proposed algorithm is the one of possible solutions of still open problem of motion estimation and prediction

References

1. Fleet, D.: *Measurement of Image Velocity*, Kluwer Academic Publishers, Massachusetts, (1992)
2. Kollios, G., Tsotras, V.: Indexing animated objects using spatiotemporal access methods, *IEEE Trans. on Knowledge and Data Engineering*, Vol. 13. 5 (2001) 758-777
3. Laplante, F.: *Real-time imaging. Theory, Techniques, and Applications*. IEEE Press, USA, (1996)

4. Chung H.Y. et al.: Fast Motion Estimation with Search Center Prediction, University of Hong Kong. www.Eee.hku.hk (2004)
5. Cheung, C.: Fast motion estimation techniques for video compression, Hong-Kong University, www.cityu.edu.hk (1998)
6. Jahne, B. Haußecker, H.: Computer Vision and Applications, Academic Press, USA (2000)
7. Lagendijk, I.: Motion estimation. information and communication theory group, Delft University of Technology, www-ict.its.tudelft.nl/ (1999)
8. Hong, T. Hongche, L.: Motion-model-based boundary extraction and a real-time implementation, *Comp. Vision and Image Understanding* Vol. 70. 1 (1998) 87-100
9. Hongche, L. Hong, T.: Accuracy vs efficiency trade-offs in optical flow algorithms, *Comp. Vision and Image Understanding*, Vol. 72. 3 (1998) 271-286
10. Smith, S. Brady, J.: SUSAN - a New Approach to Low Level Image Processing, *Int. J. of Comp. Vision*, Vol. 23. 1 (1997) 45-78
11. Lau, C.: Neural Networks, Theoretical Foundations and Analysis, IEEE Press, USA (1988)
12. Kevin, P. M.: Dynamic Bayesian Networks: Representation, Inference and Learning, Ph.D. Thesis. Computer Science. University of California, Berkeley (2002)
13. Brian, G.: Motion Compensated Noise Reduction. Graphics Research Group Microsoft Research, WA., USA (2004)
14. Sobel, J.: Machine vision for three-dimensional scenes, Academic Press, New York (1990)
15. Starostenko, O. Neme, J.A. Automatic Complex Glyphs Recognition and Interpretation, *International Journal IEICE Transaction*, Japan, Vol. E82-A. 10 (1999) 2154-2160



Filtro para la supresión de ruido impulsivo y multiplicativo para aplicaciones en imágenes médicas

José Luis Varela-Benítez¹, Francisco Gallegos-Funes¹, Volodymyr Ponomaryov²

¹ Escuela Superior de Ingeniería Mecánica y Eléctrica, U.P.A.L.M. Zacatenco
Av. IPN s/n, Col. Lindavista, C.P. 07738, Mexico, D. F.

² Escuela Superior de Ingeniería Mecánica y Eléctrica, U.P. Culhuacan
Av. Santa Ana 1000, Col. San Francisco Culhuacan, C.P. 04430, Mexico, D. F.
jlvarrel@hotmail.com, fgallegosf@ipn.mx

Resumen. En este artículo se presenta un novedoso filtro de orden estadístico para la supresión de ruido impulsivo y multiplicativo con una buena preservación de detalles para aplicaciones de procesamiento de imágenes médicas. El filtro propuesto usa la combinación del filtro L y la utilización del estimador RM (Rank M-type) en su esquema de filtrado. Para la evaluación del rendimiento del filtro propuesto fueron usados los criterios PSNR y MAE.

1 Introducción

Actualmente existen diversas clases de filtros para remover ruido en imágenes digitales [1,2]. Estos filtros se clasifican principalmente en filtros lineales y no lineales. Los filtros lineales son eficientes para remover ruido Gaussiano pero no así para ruido impulsivo. Los filtros no lineales pueden suprimir diferentes tipos de ruido incluyendo el impulsivo además de preservar contornos y detalles finos de las imágenes [1,2]. Entre los filtros no lineales se encuentran los de orden estadístico basados en la teoría de estimación robusta [3] mediante el uso los estimadores M , R y L , con los cuales se han desarrollado una gran cantidad de filtros, como son, por ejemplo, el filtro mediano, el KNN , filtro L , etc., los cuales presentan en forma individual ventajas y desventajas como son buena eficiencia en el filtrado de un tipo de ruido, pero no para todos los tipos de ruido, y problemas con la preservación de detalles y contornos. Es por eso que continúa la investigación e implementación de nuevos filtros tratando de obtener mejores eficiencias en el filtrado de mezclas complejas de ruido así como en la preservación de los detalles de la imagen [4,5].

Anteriormente, se presento el filtro $RM-KNN$ (Rank M-type K-Nearest Neighbor filter) [5], el cual usa la combinación del estimador robusto RM en el esquema de filtrado del filtro KNN . Dicho filtro provee de una buena supresión de mezclas de ruido y de preservación de detalles. En este artículo se propone una mejora del filtro L [1,4], esto es, que combine el estimador RM dentro del esquema de filtrado del filtro L , buscando obtener una mejor supresión de ruido y preservación de detalles en comparación al filtro L típico.

2 Filtro propuesto *RM-L* (*Rank M-type L-filter*)

El filtro propuesto es llamado filtro *RM-L* y fue obtenido de la combinación de los estimadores *R*, *M*, y *L*, esto es, usamos el estimador *RM* dentro del esquema de filtrado del filtro *L*.

El filtro *L* está dado por [1, 2, 4]:

$$\theta_L = \sum_{i=1}^n a_i \cdot X_{(i)} \quad (1)$$

donde $X_{(i)}$, $i = 1, \dots, n$ son las muestras de datos ordenadas y a_i , $i = 1, \dots, n$ son los coeficientes del filtro.

Por conveniencia el filtro *L*, puede escribirse de la siguiente manera:

$$\theta_L = \sum_{i=1}^n a_i \cdot \psi(x_i) \cdot x_i \quad (2)$$

donde $\psi(x_i) = \begin{cases} 1 & i = (n+1)/2 \\ 0 & \text{cualquier otro caso} \end{cases}$ es la función de influencia usada en el

filtro *L*.

El estimador *RM* provee buenas características para la supresión de ruido impulsivo y multiplicativo [5],

$$\theta_{\text{medM}} = \text{MED}\{X_i \varphi(X_i - \text{MED}\{\bar{X}\}), i = 1, \dots, N\} \quad (3)$$

donde φ es la función normalizada $\psi: \psi(X) = X\varphi(X)$ y X_i son muestras de datos.

Para mejorar las propiedades robustas del filtro *L* estándar (2), tenemos que sustituir la estimación *RM* (3) dentro del esquema de filtrado del filtro *L*,

Entonces, el filtro *RM-L* diseñado esta dado por:

$$\theta_{\text{RM-L}} = \frac{\text{MED}\{a_i \cdot [x_i \cdot \psi(x_i - \text{MED}\{\bar{X}\})], i = 1, \dots, n\}}{a_{\text{MED}}} \quad (4)$$

donde $\theta_{\text{RM-L}}$ es la salida del filtro *RM-L*, $x_i \cdot \psi(x_i - \text{MED}\{\bar{X}\})$ son los píxeles seleccionados de la ventana de filtrado por acción de la función de influencia, a_i son los coeficientes del filtro *L* y a_{MED} es el coeficiente del filtro *L* con que se obtiene la mediana.

3 Rendimiento del filtro propuesto.

El filtro propuesto fue implementado en MATLAB, utilizando una distribución de ruido exponencial para calcular los coeficientes a_i [1, 4],

$$f(x) = \frac{1}{2} e^{-x} \quad (5)$$

y la función de influencia utilizada fue la de Corte Simple [3, 5],

$$\psi(x_i - \text{MED}\{\mathcal{X}\}) = \begin{cases} 1, & |x_i - \text{MED}\{\mathcal{X}\}| \leq r \\ 0, & |x_i - \text{MED}\{\mathcal{X}\}| > r \end{cases} \quad (6)$$

donde r es un parámetro propuesto y $x_i - \text{MED}\{\mathcal{X}\}$ es el i -ésimo píxel menos la mediana de los píxeles de la ventana de filtrado.

Para evaluar el rendimiento de los filtros en la supresión de ruido vía PSNR y preservación de detalles vía MAE, los cuales se muestran a continuación [1, 2]:

$$\text{PSNR} = 10 \cdot \log \left[\frac{(255)^2}{\text{MSE}} \right] \text{ dB} \quad (7)$$

$$\text{MSE} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [e(i, j) - \hat{e}(i, j)]^2 \quad (8)$$

$$\text{MAE} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |e(i, j) - \hat{e}(i, j)| \quad (9)$$

donde PSNR es la relación pico de señal a ruido en dB, MSE es el error cuadrático medio, MAE es el error absoluto medio, MN es el tamaño de la imagen, $e(i, j)$ es la imagen original y $\hat{e}(i, j)$ es la imagen restaurada. Se realizaron pruebas al filtro propuesto usando una ventana de filtrado 3×3 en la imagen “corazón” corrompida con 5% y 20% de ruido impulsivo, así como con varianzas de 0.05 y 0.10 de ruido multiplicativo. La tabla 1 muestra los resultados obtenidos en la imagen “corazón”. La Figura 1 muestra algunos resultados visuales de las pruebas realizadas de acuerdo a la Tabla 1.

Tabla 1. Resultados comparativos PSNR en dB y MAE en la imagen *corazón* corrompida por 5% y 20% de ruido impulsivo y por 0.05 y 0.10 de varianza de ruido multiplicativo, para diferentes valores de r .

% ruido impulsivo	r	5	20	30	50	75	90	150	255
5	MAE	13.8165	6.7262	5.6533	5.6984	15.8432	9.4235	20.6890	21.1962
	PSNR	22.5823	28.4581	29.5962	29.1201	19.2366	22.5977	15.9393	15.7551
20	MAE	16.5165	9.5082	8.9710	9.1405	17.2987	15.7367	24.7890	28.2708
	PSNR	21.0771	25.5445	25.9547	25.6357	19.3164	19.3468	15.1979	14.4991
σ^2 ruido multiplicativo	r	5	20	30	50	75	90	150	255
0.05	MAE	24.1790	21.0274	21.4937	22.2598	25.8419	27.4369	27.9204	27.9072
	PSNR	18.7045	20.0132	19.9106	19.3944	18.0736	17.5947	17.4479	17.4500
0.10	MAE	27.9202	25.8324	27.0319	30.8309	36.0914	37.1095	38.0195	38.3027
	PSNR	17.5135	18.2335	17.9146	16.5381	15.4947	15.3284	15.1257	15.0679

De los resultados experimentales mostrados en la Tabla 1 y en la Figura 1 podemos observar que el rendimiento del filtro propuesto es superior al rendimiento del filtro L [4] en términos de supresión de ruido impulsivo y con preservación de detalles aceptable.

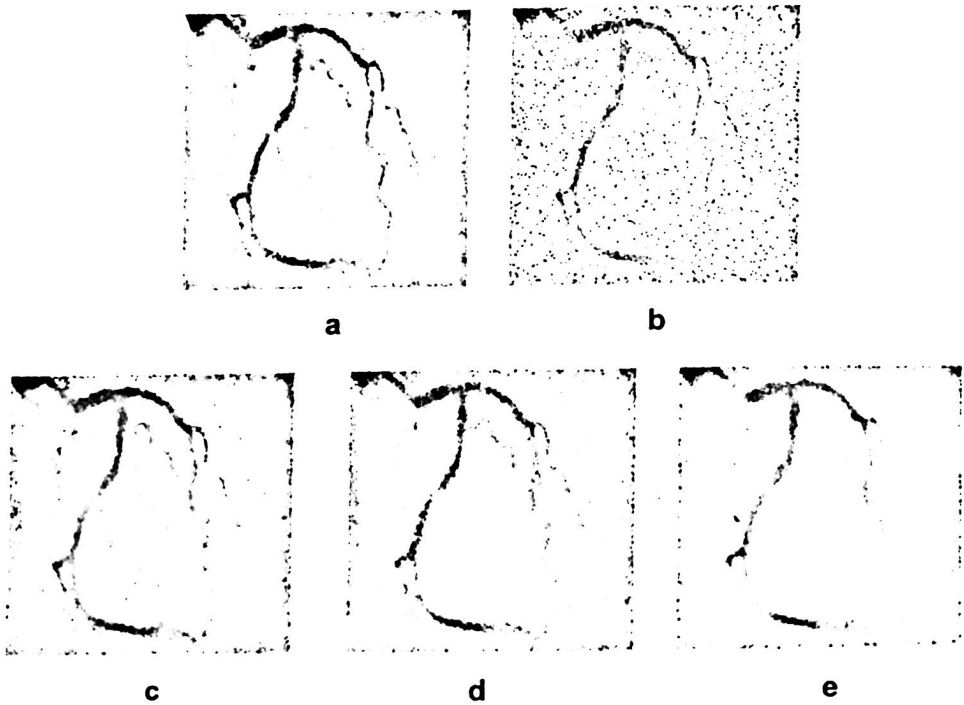


Fig. 1. Imágenes de simulación para supresión de ruido impulsivo en la imagen “Corazón” producidas por el filtro propuesto para diferentes rangos. a) Imagen original “Corazón”, b) Imagen degradada con 5% de ruido impulsivo, c) Imagen restaurada con un Rango = 30, d) Imagen restaurada con un Rango = 50, e) Imagen restaurada con el filtro L [4] (distribución Laplaciana, PNSR=29.5729).

4 Conclusiones

En la primera prueba del filtro propuesto se obtuvieron buenos resultados respecto al filtrado de ruido impulsivo, en el caso del ruido multiplicativo los resultados son aceptables aunque se tendrá que mejorar el filtro propuesto investigando otras funciones de influencia, así como la optimización de los coeficientes para otras distribuciones de ruido, por ejemplo, laplaciana, triangular, etc., y utilizando una ventana de filtrado 5×5 , esperando con ello, mejores resultados.

Agradecimientos: Los autores dan las gracias al Instituto Politécnico Nacional por los apoyos brindados para la realización de este trabajo.

Referencias

1. Pitas, A. N. Venetsanopoulos, "*Nonlinear Digital Filters*", Kluwer Academic Publishers, 1990.
2. Pitas, "*Digital Image Processing Algorithms and Applications*", John Wiley & Sons, 2000.
3. P. J. Huber, "*Robust Statistics*", John Wiley and Sons, 1981.
4. R. Öten, R. J. P. De Figueiredo, "*Sampled-Function Weighted Order Filters*", IEEE Transactions on Circuits and Systems-II: Analog and Digital Processing, Vol. 49, N° 1, January 2002.
5. F. Gallegos, V. Ponomaryov, O. Pogrebnyak, L. Niño de Rivera, "*Filtros Robustos RM-KNN con Diferentes Funciones de Influencia para Supresión de Ruido Impulsivo en Imágenes Digitales*", Computación y Sistemas, Vol. 6, No. 3 pp 183-195, 2003, CIC-IPN.

Filtro mediano con peso central iterativo en el dominio Wavelet utilizando el método de redundancia en las aproximaciones

Martínez Valdes Jesús, Gallegos Funes Francisco Javier,
Acevedo Mosqueda Marco Antonio

Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional, U.P.A.L.M. Zacatenco,
Av. IPN s/n, Col. Lindavista, C.P. 07738, México, D.F.
jesusipnmex@yahoo.com.mx, fgallegosf@ipn.mx, macevedo@ipn.mx

Resumen. En este trabajo se comparan dos métodos de supresión de ruido impulsivo en imágenes digitales a color (resolución de 24 bits) utilizando el análisis Wavelet. El primer método [1], hace uso únicamente de los coeficientes Wavelet de los filtros paso-bajo de descomposición y de reconstrucción para filtrar las muestras de la imagen ruidosa mediante una doble operación de convolución, consiguiéndose así redundar el valor de las muestras y suavizar la imagen. El segundo método a diferencia del primero consigue eliminar el ruido tanto en las bajas como en las altas frecuencias (aproximaciones y detalles de la imagen respectivamente) aplicando el primer método de filtrado antes mencionado solamente a las aproximaciones de la imagen ruidosa y un filtro basado en el análisis de los histogramas de los coeficientes Wavelet de una imagen [2], el cual se aplica tanto a las aproximaciones como a los detalles, en éstos últimos utilizando la propiedad de escalamiento propia del análisis Wavelet.

1 Introducción

El ruido es una señal constituida de componentes aleatorios tanto en amplitud como en fase, es por ello que su amplitud exacta en un instante de tiempo no puede ser predicha [4], además, debido a que el ruido se presenta en todo el espectro de frecuencias, es prácticamente imposible removerlo utilizando un solo método de filtrado.

En particular, la dificultad para suprimir el ruido impulsivo en imágenes digitales radica en la detección del pixel corrompido, el cual puede tomar valores arbitrarios dentro del rango dinámico disponible.

El primer método de filtrado que se analiza en éste trabajo da una solución práctica al problema de la supresión ruido impulsivo presente en las imágenes digitales a color sin la necesidad de encontrar un valor de umbral [1,5], además, presenta mejores resultados en comparación con el comúnmente utilizado método de umbralización [1] que también se utiliza para la eliminar este tipo de ruido. Este método de filtrado utiliza los coeficientes Wavelet de los filtros paso-bajo de descomposición y de reconstrucción para llevar a cabo una doble operación de convolución con las muestras

de la imagen ruidosa para así suavizarla y eliminar parte del ruido presente en la misma. Se hace notar que al utilizar solamente los coeficientes Wavelet de los filtros paso-bajo se pierde gran cantidad de información que concierne a los detalles finos de la imagen (información correspondiente a las altas frecuencias), pero como es bien sabido, el análisis Wavelet en el Dominio Discreto concentra la mayor parte de la energía de una señal en las bajas frecuencias y es por ello que este método de filtrado es una buena alternativa que se puede utilizar en la tarea de la supresión del ruido impulsivo.

El segundo método que se presenta utiliza la redundancia de muestras (primer método) para suprimir el ruido impulsivo presente en las bajas frecuencias y para las altas frecuencias hace uso del *filtro mediano con peso central iterativo en el dominio Wavelet* [2] que ha demostrado ser bastante efectivo en la detección del píxel ruidoso. El *filtro mediano con peso central iterativo* es una propuesta diseñada en base al análisis de los histogramas de los coeficientes Wavelet pertenecientes a los detalles de una imagen (altas frecuencias), por eso al conjuntar este resultado de supresión de ruido impulsivo en altas frecuencias con el primer método que se utilizó para el mismo fin pero para bajas frecuencias, se obtiene una supresión de ruido más completa la cual se refleja en los resultados obtenidos de las simulaciones hechas y que presentan una mejora de varios dB en los valores de relación pico señal a ruido. Se hace notar que el *filtro mediano con peso central iterativo* también se aplica a las aproximaciones de la imagen previamente filtradas utilizando el método de redundancia de muestras [1], siendo posible aplicar solamente un nivel de escalamiento a diferencia de los detalles donde se puede utilizar hasta un nivel 5 en el escalamiento, reduciéndose así la cantidad de muestras tomadas en el procesamiento de la imagen y por lo tanto el tiempo necesario para llevarse a cabo.

2 Filtro de redundancia de muestras basado en los coeficientes Wavelet de los filtros paso-bajo

Como se mencionó anteriormente este filtro toma directamente los valores de las muestras de una imagen ruidosa y los convoluciona dos veces con los coeficientes Wavelet de los filtros paso-bajo de descomposición y de reconstrucción, obteniéndose así una redundancia en el valor de las muestras y reduciendo la variabilidad de las mismas para suavizar el aspecto de la imagen y reducir el ruido. A continuación se explica paso a paso como se lleva a cabo este proceso de filtrado.

Primero se separan las muestras de la imagen ruidosa en sus distintos componentes en color como se observa en la figura 1. Donde $I_s(i, j, k)$ representa la imagen en color degradada con ruido impulsivo e $I_{SR}(i, j)$, $I_{SV}(i, j)$ e $I_{SA}(i, j)$ representan las componentes en color rojo, verde y azul respectivamente de dicha imagen.

Después, se lleva a cabo la doble operación de convolución entre las muestras de la imagen y los coeficientes Wavelet de los filtros paso-bajo de descomposición y de reconstrucción, con lo que se obtiene la imagen filtrada, estas operaciones se ilustran en las figuras 2 y 3.

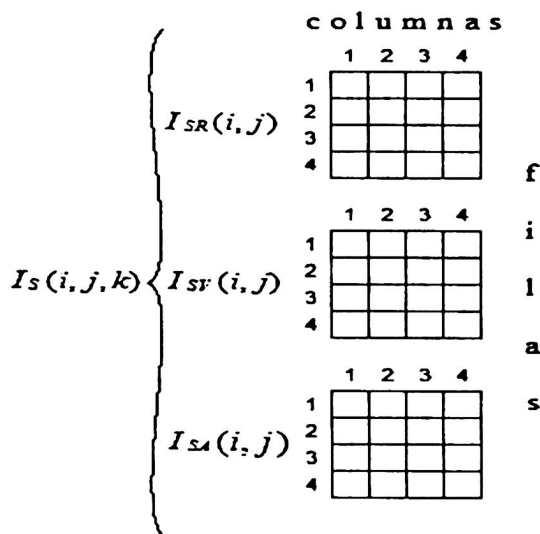


Fig. 1. Separación de una imagen ruidosa en sus distintos componentes de color.

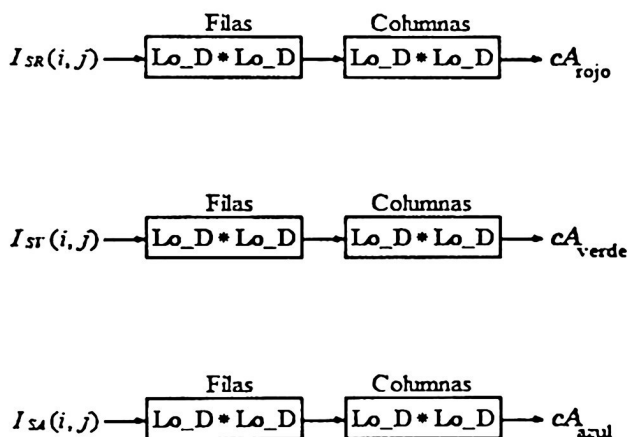


Fig. 2. Doble operación de convolución entre las muestras de la imagen ruidosa y los coeficientes Wavelet del filtro paso-bajo de descomposición.

Donde Lo_D representa los coeficientes Wavelet del filtro paso-bajo de descomposición, "*" representa la operación de convolución y cA_{rojo} , cA_{verde} y cA_{azul} representan a las aproximaciones de las matrices bidimensionales de la imagen corrompida en distintos componentes en color.

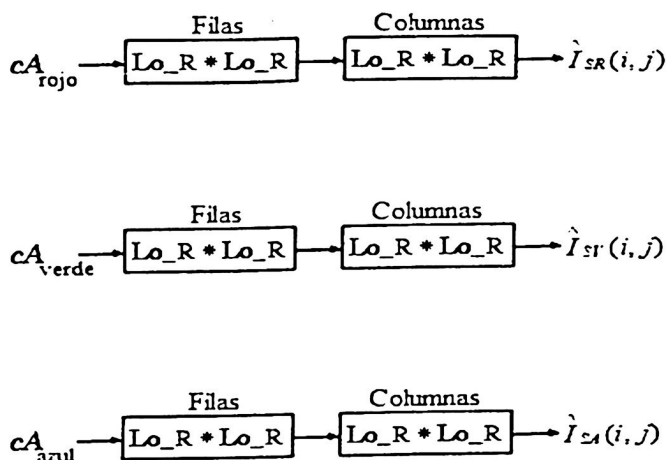


Fig. 3. Doble operación de convolución entre las aproximaciones de las matrices bidimensionales de la imagen ruidosa en distintos componentes en color y los coeficientes Wavelet del filtro paso-bajo de reconstrucción.

Donde Lo_R representa los coeficientes Wavelet del filtro paso-bajo de reconstrucción e $I_{SR}(i, j)$, $I_{SV}(i, j)$ e $I_{SA}(i, j)$ representan las componentes en color rojo, verde y azul de la imagen filtrada respectivamente. En la tabla 1 se muestran los resultados de las simulaciones realizadas para este método de filtrado.

3 Filtro mediano con peso central iterativo en el dominio Wavelet utilizando el método de redundancia en las aproximaciones

En la búsqueda de la supresión del ruido impulsivo en imágenes digitales se han utilizado métodos lineales y no lineales, éstos últimos preservando mucho mejor la información de bordes y detalles [3]. El filtro mediano es un claro ejemplo de métodos no lineales que por sus características ofrece grandes ventajas en la estimación del valor del píxel ruidoso.

En particular, este filtro mediano que se presenta en este trabajo está constituido a su vez por otro dos: el primero de ellos es el *filtro de redundancia de muestras basado en los coeficientes wavelet de los filtros paso-bajo* [1] que se trató en el punto anterior, el cual sólo repercute en la supresión de ruido en las bajas frecuencias (aproximaciones de la imagen ruidosa) y que como se muestra en el diagrama de la figura 4 es el primero en orden de aplicación, el segundo es el *filtro mediano con peso central iterativo en el dominio Wavelet* que se caracteriza por tener una gran exactitud para poder determinar si el coeficiente Wavelet de una imagen ruidosa contiene ruido o no y trabaja con las varianzas y desviaciones estándar de los mismos y no directamente con sus magnitudes (entiéndase por coeficientes Wavelet de una imagen ruidosa o coeficientes Wavelet ruidosos al resultado de convolucionar los coeficientes de

cualquier familia Wavelet en el dominio discreto con las muestras de una imagen degradada con ruido impulsivo).

Tabla 1. Resultados obtenidos de la relación pico señal a ruido entre los componentes en color de la imagen original y la imagen filtrada utilizando el filtro de redundancia de muestras basado en los coeficientes Wavelet de los filtros paso-bajo, algunos parámetros como el error de cromaticidad media y la diferencia de color normalizada son agregados para evaluar la retención de cromaticidad y cuantificar el error perceptual.

Imagen utilizada en el análisis: Lena.bmp degradada con ruido salt & pepper de 0.2%								
Wavelet	Índice de Correlación Cruzada	PSNR (dB)			Error de Cromaticidad Media			Diferencia de Color Normalizada
		Rojo	Verde	Azul	Rojo	Verde	Azul	
Harr	0.988070	46.8055	49.2783	51.6962	0.0092	0.0072	0.0056	0.203580
db2	0.984664	44.7841	46.8784	48.8817	0.0113	0.0092	0.0075	0.237174
db4	0.981436	43.1308	44.9867	46.7873	0.0133	0.0011	0.0092	0.264643
db8	0.978844	41.9641	43.6770	45.3588	0.0150	0.0126	0.0107	0.284853
coif1	0.984494	44.6912	46.7691	48.7614	0.0114	0.0093	0.0076	0.238696
coif2	0.981198	43.0181	44.8582	46.6483	0.0135	0.0112	0.0094	0.266568
coif3	0.979590	42.2863	44.0379	45.7511	0.0145	0.0122	0.0103	0.279181
coif4	0.978599	41.8594	43.5613	45.2332	0.0152	0.0128	0.0108	0.286686
coif5	0.977910	41.5727	43.2432	44.8855	0.0156	0.0132	0.0112	0.291810

El *filtro mediano con peso central iterativo en el dominio Wavelet* (ICWMF) se divide en dos etapas principalmente: en la primera etapa se lleva a cabo la detección del coeficiente Wavelet ruidoso, si el caso fue afirmativo se pasa a la segunda etapa y si el caso fue negativo se mantiene el valor de dicho coeficiente, si se pasó a la segunda etapa se lleva a cabo una nueva detección del coeficiente para diferenciar si el mismo contiene ruido o en realidad contiene información de un detalle fino de la imagen, si se determinó que el coeficiente contiene ruido se hace una estimación de su valor utilizando el valor de mediana y si se determina que el coeficiente es un detalle fino de la imagen se le da un pequeño realce para que sea más notorio. Por otro lado el escalamiento aplicado a los coeficientes Wavelet ruidosos de una imagen durante la aplicación de este filtro es determinado mediante la utilización de los histogramas de las desviaciones estándar de los coeficientes Wavelet de detalles pertenecientes a distintas imágenes sin y con ruido. Al comparar dichos histogramas se pudo apreciar que una profundidad de hasta 5 escalas puede ser utilizada para determinar un umbral o valor de discernimiento entre las desviaciones estándar mencionadas anteriormente.

El *filtro mediano con peso central iterativo en el dominio Wavelet* fue diseñado originalmente para suprimir el ruido presente en las altas frecuencias de una imagen corrompida, pero en la implementación del *filtro mediano con peso central iterativo en el dominio Wavelet utilizando el método de redundancia en las aproximaciones* se utiliza también para suprimir el ruido presente en las bajas frecuencias de la imagen (aproximaciones) con la diferencia de que el nivel de escalamiento aplicado en tales aproximaciones sólo es de la diferencia de los detalles, en donde se puede aplicar hasta un nivel de 5 escalas.

El proceso llevado a cabo durante la aplicación del *filtro mediano con peso central iterativo en el dominio Wavelet utilizando el método de redundancia en las aproximaciones* se muestra en el diagrama de la figura 4 y los resultados de las distintas simulaciones realizadas que muestran una mejora notoria de este filtro con respecto al método de filtrado presentado con anterioridad se muestran en la tabla 2.

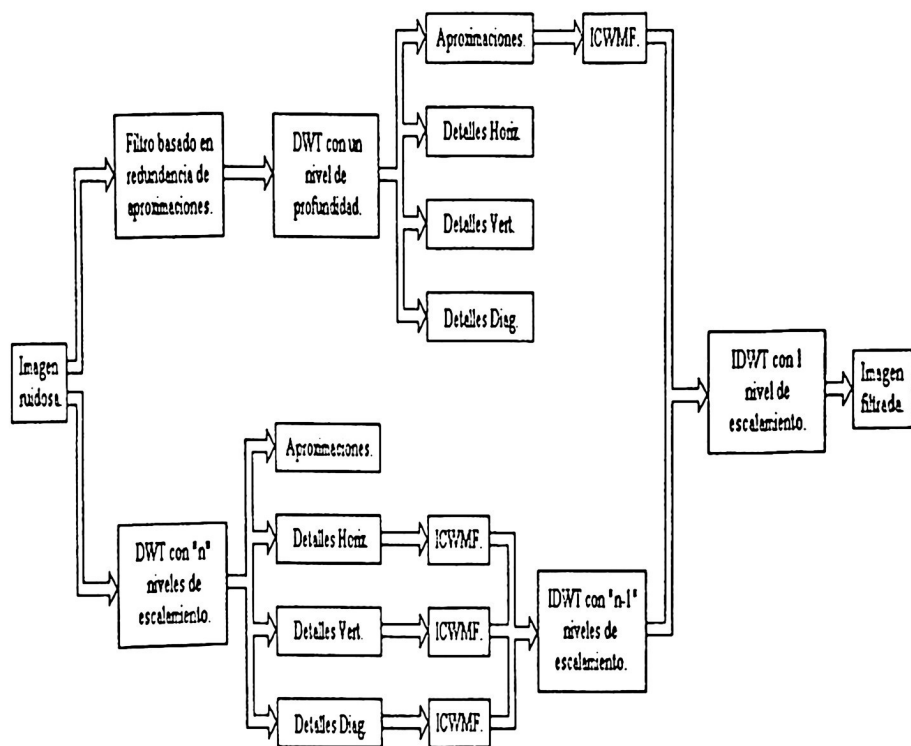


Fig. 4. Diagrama del proceso llevado a cabo durante la aplicación del *filtro mediano con peso central iterativo en el dominio Wavelet utilizando el método de redundancia en las aproximaciones*.

Tabla 2. Resultados obtenidos de la relación pico señal a ruido entre los componentes en color de la imagen original y la imagen filtrada utilizando el *filtro mediano con peso central iterativo en el dominio Wavelet* utilizando el *método de redundancia en las aproximaciones*, algunos parámetros como el error de cromaticidad media y la diferencia de color normalizada son agregados para evaluar la retención de cromaticidad y cuantificar el error perceptual.

Imagen utilizada en el análisis: Lena.bmp degradada con ruido salt & pepper de 0.2%								
Wavelet	Índice de Correlación Cruzada	PSNR (dB)			Error de Cromaticidad Media			Diferencia de Color Normalizada
		Rojo	Verde	Azul	Rojo	Verde	Azul	
Harr	0.987400	47.4537	50.5754	52.4409	0.0086	0.0063	0.0052	0.192596
db2	0.988141	48.1568	51.6351	53.0080	0.0081	0.0057	0.0049	0.182432
db4	0.987449	47.5725	50.9002	52.3651	0.0085	0.0061	0.0053	0.189214
db8	0.987260	47.3575	50.5627	51.8933	0.0087	0.0063	0.0055	0.193255
coif1	0.988326	47.8071	51.3238	52.8501	0.0083	0.0059	0.0050	0.186242
coif2	0.987425	47.5160	50.7691	52.2809	0.0086	0.0062	0.0053	0.189691
coif3	0.988710	47.9851	51.4267	52.7911	0.0082	0.0058	0.0050	0.189670
coif4	0.987306	47.2868	50.4232	51.8765	0.0088	0.0064	0.0055	0.192900
coif5	0.988769	47.8892	51.3023	52.6289	0.0083	0.0059	0.0051	0.190785

En la figuras 6 y 7 se muestran los resultados visuales obtenidos al aplicar los dos métodos de filtrado vistos en este trabajo a la imagen degradada con ruido impulsivo de la figura 5.



Fig. 5. Imagen degradada con una densidad de 0.2 de ruido impulsivo.



Fig. 6. Imagen filtrada utilizando el *método de redundancia de muestras basado en los coeficientes Wavelet de los filtros paso-bajo*.



Fig. 7. Imagen filtrada utilizando el *filtro mediano con peso central iterativo en el dominio Wavelet utilizando el método de redundancia en las aproximaciones*.

4 Conclusiones

El *filtro mediano con peso central iterativo en el dominio Wavelet utilizando el método de redundancia en las aproximaciones* tiene un mejor desempeño en la supresión de ruido impulsivo en imágenes digitales a color que el *filtro de redundancia de muestras basado en los coeficientes wavelet de los filtros paso-bajo*, esto se debe a que el primero presenta una mejora en la supresión de ruido impulsivo de varios deci-

beles, valores de índice de correlación cruzada más altos lo cual indica que el parecido entre la imagen original sin ruido y la imagen filtrada es mayor, el error de cromaticidad media en los tres componentes en color es menor lo cual indica que hay una mejor retención de cromaticidad y la diferencia de color normalizada también es menor lo cual indica que un observador percibe menos errores de color en la imagen filtrada.

Agradecimientos: Los autores agradecen al Instituto Politécnico Nacional el apoyo brindado para llevar a cabo este trabajo.

Referencias

1. Jesús Martínez Valdes, Francisco J. Gallegos Funes, Marco Antonio Acevedo Mosqueda, *Reduccion de Ruido Impulsivo en Imágenes Digitales a Color Utilizando Wavelets*, Memorias de IEEE 15 Reunión de Otoño de Comunicación y Exposición Industrial ROC&C 2004, Acapulco, México, Noviembre 2004.
2. S. M. Mahbubur Rahman, Md. Kamrul Hasan, *Wavelet-domain iterative center weighted median filter for image denoising*, Signal Processing, 83, 1001-1012, 2003.
3. S. K. Mitra and G. L. Sicuranza, *Nonlinear Image Processing*, Academic Press, San Diego, CA, 2001.
4. C. D. Motchenbacher, *Low-Noise Electronic System Design*, John Wiley & Sons, Inc., New York, 1993.
5. D. L. Donoho, *Denoising by wavelet thresholding*, IEEE Trans. Inform. Theory, 41(3) pp. 613-627, 1995.

Sistema de acceso al personal mediante la autenticación del iris

Juan José Torres Manríquez¹, Edmundo René Durán Camarillo¹,
Carlos Alberto González Dueñas², Aranzazú del Carmen González Montiel²,
Verónica Martínez Mora², Carlos Vidal Mendoza González²

¹Escuela Superior de Cómputo. Departamento de Posgrado. IPN. Av. Juan de Dios Bátiz Esq. Miguel Othón de Mendizábal S/N, Unidad Profesional Adolfo López Mateos, Col. San Pedro Zacatenco. México D.F., C.P. 07738. Tel. 5729.6000 ext. (52021).

{jtorresm, eduranc}@ipn.mx

²Escuela Superior de Cómputo. Á

rea de Trabajo Terminal. IPN. Av. Juan de Dios Bátiz Esq. Miguel Othón de Mendizábal S/N, Unidad Profesional Adolfo López Mateos, Col. San Pedro Zacatenco. México D.F., C.P. 07738.
{carlos_gduenas, aranza_glez, yousve, vidalipn}@hotmail.com

Resumen. El proyecto está inspirado en los sistemas de alta seguridad basados en índices biométricos. Los patrones dentro de la textura del iris, ofrecen un medio de identificación confiable, debido a que se trata de un patrón único en cada persona, se puede considerar la textura del iris como una "huella óptica inalterable". El proyecto consiste de un sistema de cómputo para la autenticación de personas a través de la textura del iris basado en algoritmos de procesamiento de imágenes y redes neuronales artificiales. El sistema está integrado por un módulo de captura y una interfaz gráfica. El proceso que el sistema realiza se describe brevemente de la siguiente manera: El módulo de captura, capta la imagen del ojo del individuo, la lleva a través de una interfaz a una computadora personal y dicha imagen es objeto de un procesamiento de imágenes para la detección del iris y generación de un código único. Una vez que se ha generado y almacenado el código del individuo, éste es comparado con una base de conocimiento, la cual autenticará al individuo (siempre y cuando esté registrado), dando acceso al personal a un área restringida.

1 Introducción

En el ámbito de las tecnologías de seguridad, uno de los problemas fundamentales a resolver es la necesidad de autenticar de forma segura la identidad de las personas que pretenden acceder a un determinado servicio o recinto físico, por ello se han creado diferentes sistemas que se han auxiliado de las técnicas de identificación biométrica, con el objetivo de resolver este problema a partir de características que son propias de cada persona, como su voz, huella dactilar, iris, rostro, etc. La medición biométrica se ha venido estudiando desde tiempo atrás y es considerada en la actualidad como uno de los mejores métodos de identificación humana.

Los sistemas biométricos incluyen un dispositivo de captura y un software biométrico que interpreta la muestra física y la transforma en una secuencia numérica. Entrar en un lugar físico mediante el reconocimiento del iris resultará bastante práctico para los usuarios, ya que además de no tener que portar tarjetas o llaves tampoco hay que recordar largas contraseñas.

Utilizar el iris como clave de acceso en un sistema puede resultar muy seguro, ya que éste cuenta con 266 características que pueden ser medidas y no hay dos personas en todo el mundo con el mismo iris; de hecho, el iris derecho es diferente al izquierdo. Además, el iris no puede ser manipulado, copiado ni robado. Por otro lado, las características del iris son muy complejas, lo cual proporciona un modelo muy preciso para la autenticación de cada persona, teniendo una tasa de error de entre 1 y 1.2 entre un millón, desde el punto de vista biométrico [1].

Existen diferentes empresas que se han dedicado a la creación de sistemas que autentican a las personas mediante el patrón biométrico del iris, el origen de dichas empresas son generalmente de diferentes partes del mundo (excluyendo a México). La empresa estadounidense IriScan es la principal desarrolladora de tecnología basada en reconocimiento de iris que existe actualmente, ya que posee la patente sobre esta tecnología [2].

El presente proyecto desarrollará un sistema propio para dar una aplicación específica de acuerdo a las necesidades que se han detectado, en este caso se propone un prototipo que en el futuro se aplique en cuneros de hospitales.

2 Desarrollo

En el diagrama a bloques de la Fig. 1, se estructura de manera general el proceso que se ha planteado para desarrollar el sistema.

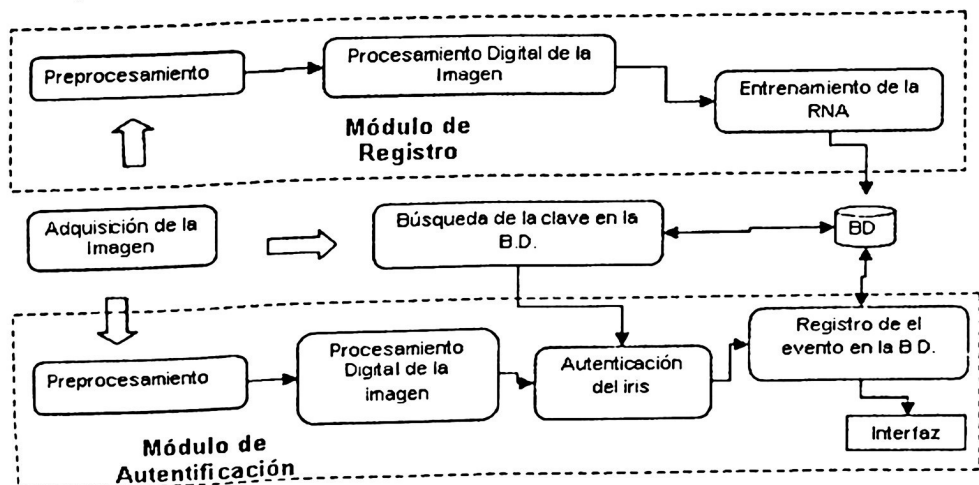


Fig. 1. Diagrama a Bloques del Sistema.

2.1 Módulo de adquisición de imagen

En la Fig. 2 se puede observar que el módulo de adquisición de imagen está constituido por un sistema de dos puertas, en la primera se tiene un teclado matricial en el cual los usuarios teclearán una pequeña clave, dicha información pasará a la computadora con ayuda de un microcontrolador, en la computadora la clave se usará para verificar la existencia del usuario en la Base de Datos, de ser así la puerta se abrirá, posteriormente pasará al cubo donde el usuario tendrá que posicionarse frente a la cámara y presionar el botón donde se tomará la fotografía, enviándola a la computadora por un cable USB donde se almacenará la imagen para ser preprocesada, procesada y posteriormente pasará al módulo de autenticación, si el resultado arrojado es favorable se accionará la segunda puerta tomando una fotografía desde la parte superior del cubo para una mejor referencia del personal que entra.

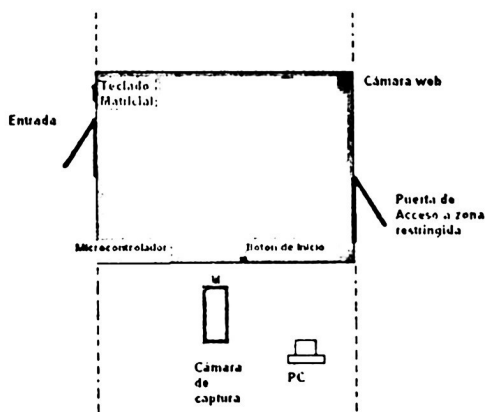


Fig. 2. Módulo de Adquisición de la Imagen.

2.2 Módulo de registro

En el Módulo de Registro el usuario enviará como entrada al sistema la imagen del iris, posteriormente ésta será enviada a la etapa de Preprocesamiento, donde será sometida a diferentes algoritmos de tratamiento digital de imágenes, obteniendo de ella el área de interés (iris) para poder enviarla a la etapa de Procesamiento, donde a través de diferentes técnicas matemáticas y en base al análisis de textura se extraerán las características del iris, dichas características serán enviadas a la etapa de Entrenamiento de la Red Neuronal Artificial, donde se obtendrá un vector de pesos para cada individuo, como resultado del entrenamiento, dicho vector será guardado en la Base de Datos para su consulta continua.

2.2.1 Preprocesamiento

La etapa de preprocesamiento tendrá como entrada a la imagen digital del iris tomada en el Módulo de Adquisición de la Imagen, este módulo será el encargado de realizar una mejora en la imagen contemplando las técnicas de eliminación de ruido, mejora de contraste, así como resaltar los detalles de interés de la imagen y descartar las partes que no sean necesarias para su análisis. Todo ello con la finalidad de aumentar las posibilidades de éxito en los procesos posteriores. En la Fig. 3 se muestra la imagen obtenida por el dispositivo de captura del sistema.

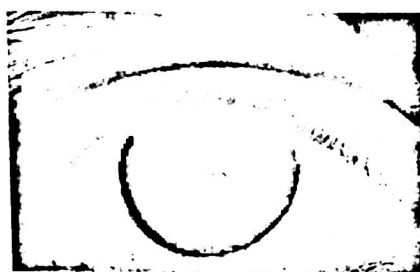


Fig. 3. Imagen obtenida por el dispositivo de captura.

Para ubicar el área de interés (iris) y eliminar el resto de la imagen que no es importante, es decir para realizar la segmentación de la misma, se aplicaron los siguientes algoritmos:

- *Clasificador K-NN*. Existen tres clases principales en la imagen (piel, esclerótica, iris), para poder definir cada clase se toman muestras de cada una de ellas en base a su color: Todo ello con el propósito de separar las regiones de dichas clases y tomar la de interés.
- *Etiquetado*. Se etiqueta la imagen en base a la conectividad-8 para definir los objetos existentes en la imagen y sus respectivas áreas, dejando el área más grande que en este caso será el objeto que involucra al iris y eliminando los objetos más pequeños. Para eliminar el ruido en el centro de la pupila se aplica un etiquetado en el fondo de la imagen y ahora se eliminan los objetos de fondo más pequeños.
- *Transformada de Distancia (City-block)*. El siguiente paso consiste en encontrar la coordenada más aproximada al centro del iris, para ello se utiliza la transformada de distancia, es decir, se va obteniendo en cada píxel la distancia de dicho punto al fondo dejando únicamente la mayor distancia. Debido a la morfología del iris, la transformada de distancia no da el mejor resultado ya que devuelve una coordenada muy alejada del centro. Para resolver este problema se utilizan los conocimientos de geometría, considerando tres puntos que forman parte de cualquier circunferencia se sabe que se puede obtener el centro mediante la intersección de las rectas perpendiculares a las rectas que unen cada uno de los tres puntos (triángulo), esta intersección es la coordenada más cercana al centro del círculo que aproxima al iris.
- *Algoritmo de círculo de punto medio de Bresenham*. Con una aproximación del centro del iris, se implementa un generador de círculos para el recorte de la imagen

conocido como el Algoritmo de círculo de punto medio de Bresenham, para construir dichos círculos, se requiere del centro aproximado para obtener los distintos radios y adquirir la dona mostrada en la Fig. 4.

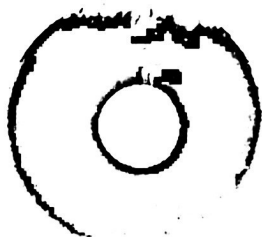


Fig. 4. Obtención de la "dona".

Para la extracción de características es necesario tomar 60° aproximadamente del área izquierda de la dona (debido a que es un área sin ruido), una vez identificada la zona de análisis se realiza la transformación de dona a cinta; utilizando nuevamente el algoritmo de Bresenham, recorriendo la imagen, partiendo de la mitad hacia arriba 30° y de igual forma hacia abajo. Obteniendo una imagen rectangular de fácil manipulación, mostrada en la Fig. 5.

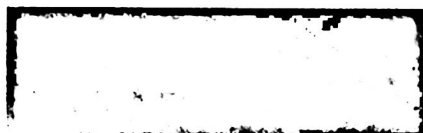


Fig. 5. Obtención de la Cintilla.

2.2.2 Procesamiento

2.2.2.1 Extracción de características y construcción del código del iris

Los Filtros de Gabor en 2D tienen la particularidad de ser muy utilizados para la extracción de características en texturas. Una estrategia para extraer información de la textura del iris, es por medio de los coeficientes de Gabor en 2D.

Estos filtros son implementados con las funciones elementales de Gabor, las cuales están constituidas por una exponencial compleja y una función gaussiana. Sus propiedades matemáticas fueron elaboradas en 1985 por su autor Daugman, quien señaló que estos filtros de dos dimensiones eran óptimos para proporcionar tanto la máxima resolución posible con información sobre la orientación y frecuencia espacial contenida en estructuras locales de una imagen y simultáneamente la información sobre la posición en dos dimensiones [3].

Éstas funciones están determinadas por cuatro parámetros, dos que expresan su localización en el dominio espacial (x, y), es decir las coordenadas del centro de la en-

volvente Gaussiana y otros dos que expresan la frecuencia espacial de sintonía F y la orientación ϕ .

En el sistema propuesto, la extracción de características es realizada utilizando un conjunto de funciones de Gabor 2D con 8 ángulos y 5 frecuencias espaciales diferentes.

La función de Gabor en 2D está dado por la Eq. (1).

$$h(x, y) = g(x', y') e^{2\pi j(Ux + Vy)} \quad (1)$$

donde $g(x, y)$ es una función Gaussiana en 2D dada por la Eq. (2).

$$g(x', y') = \left(\frac{1}{2\pi\lambda\sigma^2} \right) \exp \left[-\frac{(x'/\lambda)^2 + y'^2}{2\sigma^2} \right] \quad (2)$$

donde σ es el ancho de banda radial. Para evaluar el sistema propuesto, se utilizó un valor de $\sigma^2 = 8$. λ es un vector de forma Gaussiana, el cual se estableció como $\lambda = 1$ para obtener una forma circular. (x', y') es la posición rotada de (x, y) por el ángulo ϕ como se muestra en la Eq. (3).

$$(x', y') = (x \cos \phi + y \sin \phi, -x \sin \phi + y \cos \phi) \quad (3)$$

Las fases de modulación ϕ utilizadas en el sistema son: $0, \pi/8, \pi/4, 3\pi/8, \pi/2, 5\pi/8, 3\pi/4$ y $7\pi/8$. Para la frecuencia espacial F , se probó el siguiente grupo de frecuencias $F = 3\pi/4, 3\pi/4, \sqrt{2}, 3\pi/8, 3\pi/8\sqrt{2}, 3\pi/16$ [4].

La función de Gabor dada por la Eq. (1) puede ser descompuesta en parte real parte imaginaria como se muestra en la Eq. (4).

$$h(x, y) = h_c(x, y) - j h_s(x, y) \quad (4)$$

Donde:

$$h_c(x, y) = g(x', y') \cos(2\pi F x') \quad (5)$$

y

$$h_s(x, y) = g(x', y') \sin(2\pi F x') \quad (6)$$

h_c es una función de Gabor simétrica (función par) y h_s es una función de Gabor asimétrica (función impar). La información obtenida por estas dos funciones (5) y (6) para un ángulo específico ϕ y frecuencia espacial específica F en un punto central (x_0, y_0) permite obtener un vector bidimensional cuya magnitud corresponde a la energía de la función de Gabor específica representada por la Eq. (7).

$$M(x, y) = \sqrt{h_c^2 + h_s^2} \quad (7)$$

La función $M(x, y)$ presenta una gran similitud con el comportamiento de las células complejas de un sistema visual humano y provee una unidad de la respuesta del canal de la función específica de Gabor.

En el sistema propuesto, el número de funciones de Gabor utilizadas para cada campo receptivo es de 40.

En primer lugar la imagen de la cintilla de iris cuyo tamaño es fijo (64 x 256 pixeles) es dividida por 64 campos receptivos (4 x 16 bloques) de 16 x 16 pixeles cada uno, centrados por un punto (x_0, y_0), como se muestra en la Fig. 6. Siendo el número de campos receptivos constante e independiente del tamaño de la imagen de la cintilla..



Fig. 6. Imagen del iris segmentada por 64 campos receptivos con cada punto central.

A cada campo receptivo de la imagen de la cintilla de iris, le son aplicados 40 filtros de Gabor, resultando 2,560 valores para el total de las imágenes. (40 funciones x 64 campos receptivos).

Cuando promediamos la Eq. (7) por el número filtros de Gabor utilizados, obtenemos un vector de características como se muestra en la Eq. (8).

$$M = \frac{\sum_{p=1}^{40} M_p(x, y)}{40} \quad (8)$$

Y de esta forma se genera el código del iris compuesto por 64 coeficientes representativos, que a su vez son normalizados para obtener valores entre 0 y 1, mismos que son el vector de entrada de la Red Neuronal Artificial para la autenticación de personas. En la Fig. 7 se puede apreciar la gráfica de la magnitud de cada coeficiente del Filtro de Gabor.



Fig. 7. Gráfica de coeficientes del Filtro de Gabor.

2.2.3 Entrenamiento de la RNA

El tipo de red neuronal artificial a implementar es una red con aprendizaje supervisado, el algoritmo de aprendizaje de la Backpropagation para redes multicapa, ya que la red multicapa es una red que puede implementarse para resolver problemas de clasificación de patrones. Una red de estas características esta en capacidad de generar re-

giones de decisión arbitrariamente complejas, aunque en ciertos problemas se puede simplificar el aprendizaje aumentando capas ocultas, la tendencia es aumentar la extensión de la función de activación, en lugar de la complejidad de la red [5].

La red neuronal multicapa de tipo Backpropagation trabaja bajo aprendizaje supervisado y por tanto necesita un conjunto de entrenamiento que le describa cada entrada y su valor de salida esperado.

Cada patrón de entrenamiento se propaga a través de la red y sus parámetros para producir una respuesta en la capa de salida, la cual se compara con los patrones objetivo o salidas deseadas para calcular el error en el aprendizaje, este error marca el camino mas adecuado para la actualización de los pesos y ganancias que al final del entrenamiento producirán una respuesta satisfactoria a todos los patrones de entrenamiento, esto se logra minimizando el error medio cuadrático en cada iteración del proceso de aprendizaje. Al término del entrenamiento, los pesos obtenidos serán el código único del usuario.

La Red Neuronal recibirá como entrada el código del iris ya descrito, dicho vector en conjunto con otros son obtenidos de 5 a 10 imágenes más de la misma persona, los cuales formarán parte del entrenamiento. Durante el entrenamiento los vectores de entrada harán un ajuste de pesos por cada neurona contenidas en diferentes capas, se sabe que los vectores de entrada son semejantes debido a las distintas imágenes, ya que al tomar un número determinado de imágenes aún siendo de la misma persona muy probablemente cause cierto margen de error en cuanto a la posición o algunos factores externos propios de la persona.

2.3 Módulo de autenticación

En este módulo el usuario introduce su clave para ser consultada en la Base de Datos, si el usuario no se encuentra registrado el sistema mandará un mensaje de error, de lo contrario regresará el vector de pesos correspondiente a la clave introducida pasando al Módulo de Autenticación del Iris, en este módulo al igual que el anterior introducirá el usuario al sistema la imagen del iris y se llevará a cabo el mismo preprocesamiento y procesamiento de la imagen.

2.3.1 Autenticación de iris

Después del preprocesamiento y procesamiento, el sistema buscará la clave en la base de datos, si la clave no existe el sistema mandará un mensaje de error, de lo contrario obtendrá las características de la persona que serán comparadas con el resultado que se obtenga al recargar en la Red Neuronal Artificial el vector de pesos ya antes obtenido, para poder autenticar la identidad del usuario corroborando su existencia en la Base de Datos, de ser así el sistema enviará al usuario como respuesta la autorización al área restringida mediante la Interfaz del Sistema.

2.4 Interfaz gráfica

La interfaz gráfica del sistema, fue hecha en un entorno de “desarrollo rápido de aplicaciones” (RAD, por sus siglas en inglés), el cual contará con un compilador de lenguaje orientado a objetos, como lo es C++. Por lo que el entorno escogido para la realización del sistema fue Borland C++ Builder, ya que cumplía con las características ya mencionadas.

Esta interfaz gráfica fue realizada en base a un análisis realizado para el Hospital Infantil de México, en el cual es necesario dar de alta un administrador por turno, así como poder almacenar los siguientes datos: nombre, dirección, teléfono particular, teléfono celular, turno en el que labora, puesto que ocupa y su fotografía.

De igual forma, se sabe que son menos de cincuenta empleados los que pueden acceder al cunero, por lo que la base de datos y el sistema ha sido diseñada para admitir hasta 99 personas, teniendo en cuenta que se les asignará un id de cuatro dígitos. Para los usuarios se requiere que se guarden los mismos datos de los administradores, pero también requieren saber a que hora accedieron al cunero, para poder consultarlo tanto por nombre de empleado o fecha.

Debido a los requerimientos mencionados anteriormente se ha diseñado la interfaz gráfica. En la Fig. 8 podemos ver la pantalla principal del sistema. Para poder realizar cualquier acción en el mismo es necesario que se inicie sesión para lo cual se tiene que ingresar un nombre de usuario y una contraseña.

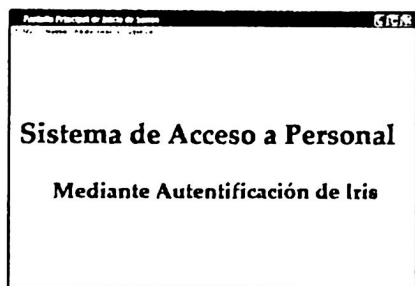


Fig. 8. Pantalla Principal.

Una vez iniciada la sesión de administrador se pueden dar de alta los usuarios, bajas o modificación de sus datos.

Para las consultas de accesos al cunero se implementaron tres pantallas diferentes:

Búsqueda por nombre. Escribiendo el nombre completo de la persona y un intervalo de fechas, nos genera un reporte de los registros que cumplan con los criterios especificados, teniendo la posibilidad de poder ver los detalles de un registro en particular.

Búsqueda por fecha. El sistema tiene la posibilidad de mostrar todas las personas que entraron en una determinada fecha, y de igual forma que en la consulta anterior podemos ver los detalles de un registro en particular. En caso de que se necesite un reporte de los usuarios que accedieron al cunero en un intervalo de fechas se puede obtener con dicho intervalo, teniendo la posibilidad de imprimir dicho reporte.

3 Conclusiones

El iris como indicador biométrico para la identificación de individuos en condiciones reales y múltiples aplicaciones de seguridad presenta un gran potencial debido a sus características y la forma de capturar las imágenes para el proceso de autenticación que excluye totalmente procedimientos invasivos para el usuario.

La caracterización del iris para sistema de autenticación de personas puede contener procedimientos robustos en sus etapas de preprocesamiento y procesamiento, lo que conlleva a considerar ciertas características introducidas por las condiciones reales tales como, efecto de párpados y pestañas, irregular ubicación del iris en la imagen, efectos producidos por la iluminación y poca área efectiva de información en la imagen. Para ello se realizaron pruebas con diferentes algoritmos de Tratamiento Digital de Imágenes y Geometría. Por lo que, después de experimentar con dichos algoritmos, los resultados más óptimos para la segmentación del Iris en las imágenes adquiridas fueron los presentado en este artículo.

En la etapa de procesamiento se utilizan los filtros de Gabor para su aplicación sobre texturas de iris, de acuerdo con la investigación que se llevo a cabo, estos filtros presentan una adecuada representación del conjunto espacio-frecuencia para este tipo de análisis, y los resultados obtenidos en problemas son muy satisfactorios, pero no se descarta otro tipo de métodos de análisis de texturas y de frecuencias espacialmente localizadas que puedan presentar mejores desempeños. Con respecto a los clasificadores, se utiliza una Red Neuronal Multicapa con algoritmo de aprendizaje Backpropagation, debido a que es una red que puede implementarse para resolver problemas de clasificación de patrones no linealmente separables, permitiendo cierta tolerancia a fallos, es decir la red será capaz de recibir nuevas entradas con ruido o incompletas dando una salida activa si la nueva entrada es parecida a las presentadas durante el aprendizaje.

Referencias

1. Iridian Technologies, Inc: www.iriscan.com
2. John G. Daugman : Recognizing Persons by Their Iris Patterns, Cambridge University.
3. John Daughman, Cathryn Downing : Gabor Wavelets for Statiscal Pattern Recognition.Part III: Articles
4. MARIKO Nakano, PÉREZ Héctor, Artículo: Automatic Face Recognition System using the Gabor Transform. México, D.F. ESIME Culhuacán IPN.
5. Martín T. Hagan (1996): Neural Network Design, Ed. PSW Publishing Company.

Natural Language Processing

Procesamiento de lenguaje natural

Agentes móviles para extractos de documentos

Lorena Leal Bando, Darnes Vilariño Ayala, Fabiola López y López

Benemérita Universidad Autónoma de Puebla, FCC

14 Sur y Av. San Claudio

lore_lb33@hotmail.com, darnes@cs.buap.mx, fabiola.lopez@siu.buap.mx

Abstract. La Recuperación de Información (RI) busca extractos o documentos que brinden una respuesta más exacta a la pregunta formulada. Con el surgimiento de Internet se ha acrecentado la necesidad de utilizar de manera efectiva la información disponible en diferentes servidores remotos. Por lo cual se requiere incorporar mecanismos eficientes de búsqueda de información distribuida espacialmente, esto incrementa el tráfico en la red. Uno de los beneficios de la tecnología de agentes móviles es precisamente, la reducción de costos en la comunicación entre los múltiples host que se pueden visitar. El objetivo de este trabajo es demostrar cómo el uso de agentes móviles agiliza la recuperación de información para lo cual usamos como herramienta a ProActive. Se proponen dos diferentes estrategias de migración.

1 Introducción

La RI es un área de las Ciencias de la Computación de gran importancia por dedicarse al estudio de sistemas y técnicas para asignar índices, buscar y devolver datos valiosos a los usuarios. En los últimos años se ha acrecentado la necesidad de utilizar de manera efectiva la información disponible en una red, como puede ser Internet [9]. Actualmente se estima que diariamente se crean 4500 sitios nuevos y sólo 350 de ellos desaparecen. En países como el nuestro, Internet crece a pasos agigantados, considerando que es una tecnología nueva que hizo su aparición hace menos de 15 años. Este crecimiento se ve reflejado en el número de usuarios, hace 3 años sólo el 3% de la población tenía acceso a Internet, ahora se estima que para el año 2005 este porcentaje se incrementará en un 13%, lo que representa 17 millones de usuarios en México, expresado por Rodolfo Sandoval, Director de Intel en México [1, 9, 13]. Otra razón por la que el tratamiento de la información se vuelve más complejo es debido a la calidad de la misma, ya que en la red se cuenta con diversos tipos de datos, que incluyen texto, imágenes, sonido, etc.

Debido a que la información está espacialmente distribuida (en sitios conectados a Internet), el problema radica en localizar efectivamente los lugares donde se pudiera encontrar información interesante. De ahí la necesidad de incorporar a la RI mecanismos que le permitan acceder a sitios remotos. Una solución inmediata sería el traer los documentos al procesador central para ser analizados. Esto incrementaría enormemente el tráfico en la red porque se tendrían que enviar cualquier documento aún cuando no fuese relevante a la solución.

3 Agentes móviles

Un agente es algo que actúa, (agente proviene del Latín *agere* que significa, hacer) [20]. Informalmente un agente es una entidad con objetivos, que alcanza ejecutando una serie de acciones, mediante un dominio de conocimiento y situado en un entorno concreto [6]. Es difícil encontrar una definición exacta para el término “agente” que incluya todas las propiedades y características que los investigadores y diseñadores consideren relevantes, es por ello que existen numerosas definiciones de agentes. A continuación se citan algunas de las más solicitadas dentro de la literatura:

Para Russel y Norving un agente es una entidad que percibe y actúa sobre un entorno a través de efectores [20]. Wooldridge y Jennings: un agente es un sistema informático situado en un entorno y que es capaz de realizar acciones de forma autónoma para conseguir sus objetivos de diseño [23].

Actualmente el paradigma de agentes ha robado la atención de varios investigadores, y su colaboración ha proporcionado gran cantidad de literatura al respecto, incluyendo definiciones, clasificaciones, diseño de arquitecturas, lenguajes y protocolos de comunicación. Existen también organizaciones como la FIPA [22, 26] y la OMG MASIF [22, 27] que como primer paso trabajan en la estandarización de sistemas de agentes, a su vez de organizar y realizar eventos para congregarse a más personas interesadas en el tema. Gracias a estas aportaciones se han desarrollado ya gran cantidad de agentes, que han sido utilizados en control de procesos, comercio electrónico, herramientas de desarrollo, software de interfaces de usuario, manejo de redes y tareas de recuperación de información.

Un agente es caracterizado por una serie de calificativos, los cuales denotan ciertas propiedades a cumplir [20, 23].

- Autonomía, el agente percibe su entorno y puede operar por sí sólo sin la intervención de humanos u otros agentes.
- Reactividad, el agente es capaz de responder a cambios en el entorno en el que se encuentra situado.
- Pro-actividad, a su vez el agente debe ser capaz de intentar cumplir sus propios planes u objetivos.
- Sociabilidad, debe de poder comunicarse con otros agentes mediante algún tipo de lenguaje de comunicación de agentes.

En la literatura se suelen atribuir otras características a los agentes en mayor o menor grado para resolver problemas particulares, una de ellas es la movilidad que se define como la capacidad de un agente de trasladarse a través de una red.

Los agentes móviles son procesos computacionales capaces de navegar por redes [4,8,11,12,13,14,15,16,18,22,25] WAN, como Internet, interactuando con equipos, acumulando información en nombre de sus propietarios, y volver después de haber realizado las tareas requeridas por el usuario para informar a este de los resultados. Los atributos de movilidad en los agentes han introducido el concepto de programación remota, donde un mismo agente puede actuar tanto de cliente como de servidor. Este tipo de agentes tienen diversas características que los hacen atractivos los desarrolladores ya que deben ser portables a través de plataformas, capaces de

elegir cuando y donde transportarse a sí mismo, duplicarse, comunicarse con otros agentes para intercambiar información [4,11,12].

Los agentes móviles agregan dos cosas:

- Datos (datos recopilados y estados de procesos) y;
- Código (instrucciones que dirigen la conducta).

Un agente móvil debe ser hábil para ejecutarse en cualquier máquina dentro de una red, independiente del tipo de procesador o sistema operativo. El código del agente no debe haber sido instalado en cada máquina que el agente pudiera visitar potencialmente; este debe moverse con datos del agente [11].

Son ideales por que se reduce la carga de red al evitar el movimiento de datos, se mueven los cálculos hacia los datos y no al revés, se descentraliza los procesos al asignar tareas específicas en diferentes lugares, permiten no estar conectados a la red, mejoran la latencia de red en sistemas de tiempo real y para redes de gran tamaño [12, 14,16].

Por el contrario, tienen una serie de riesgos relativos a la seguridad, la confidencialidad y la autenticación. En primer lugar un agente es representante de un usuario, por consiguiente, se debe garantizar y probar en todo momento la identidad del usuario a quien representa evitando suplantaciones fraudulentas. Además, los agentes no actúan en solitario, sino que se localizan en entornos o plataformas con varios agentes, donde cada uno tiene sus propios objetivos, toma sus decisiones y puede tener la capacidad de comunicarse con otros *agentes*, lo que crea nuevos riesgos en las comunicaciones. Dichos entornos se conocen como sistemas multi-agente [12, 14, 16, 18].

4 ProActive

Para poder desarrollar agentes móviles y que estos puedan migrar entre varios hosts, es preciso utilizar plataformas específicas destinadas a tal fin, que permitan a los agentes móviles desplazarse entre las máquinas que componen dicha plataforma. La plataforma debe proporcionar facilidades para el desarrollo del agente móvil, tales como el soporte donde implementar su modelo de ciclo de vida (creación, clonación y migración). Además debe facilitar la comunicación entre agentes (o entre agentes y un usuario u otra entidad) y proporcionar servicios de seguridad para evitar que agentes maliciosos causen estragos.

ProActive es una API que permite desarrollar aplicaciones formadas por varios subsistemas y cada uno de ellos compuesto por Objetos Activos (OA). En ProActive los OA's son unidades básicas de actividad y distribución para aplicaciones concurrentes [28].

El OA está compuesto por dos objetos: un *cuerpo* y un objeto Java estándar. El cuerpo no es visible desde el exterior del objeto activo, y todo luce como si el objeto estándar fuese activo. El cuerpo es responsable de recibir los llamados del OA, los almacena en una cola de peticiones y posteriormente ejecuta estos llamados en un orden específico según la política de sincronización establecida [2, 14, 28].

Para evitar esto se propone el uso de agentes móviles que puedan procesar localmente la información y sólo transmitan aquella que sea relevante [5, 7, 10]. Los agentes móviles son programas que pueden migrar de un host a otro en una red. El objetivo de este trabajo es demostrar cómo el uso de agentes móviles agiliza el tiempo de búsqueda de información, y a la vez permite que sea un proceso menos tedioso para los usuarios. En nuestra propuesta, los agentes migran a varios hosts usando diferentes estrategias de migración, aplican técnicas de RI y regresan sólo con aquella información que sea relevante a la consulta solicitada. Para ello usamos ProActive [28] que es una librería pura de Java que permite realizar aplicaciones paralelas, distribuidas y concurrentes. ProActive nació de un proyecto del Consorcio ObjectWeb y del Instituto Nacional de Investigación en Informática y en Automática (INRIA) en Francia. El artículo está organizado de la siguiente manera. Las secciones 2, 3 y 4 proporcionan los fundamentos teóricos de la extracción de información, conceptos relacionados a los agentes móviles y una descripción general de la plataforma ProActive, respectivamente. La sección 5 da al lector la visión general de la implementación de este trabajo. Por último, en las secciones 6 y 7 se exponen los resultados obtenidos hasta el momento y las conclusiones del mismo, respectivamente.

2 Extractos de documentos

El tesoro más valioso del hombre es la información y sin lugar a dudas el potencial de este tesoro radica en la habilidad para realizar ciertas operaciones con la información. Para procesar una gran cantidad de datos se ha hecho uso de la computadora, pero lo que se tiene que considerar es que no son capaces de entender los textos para su procesamiento, para una computadora un documento sólo representa cadenas de caracteres sin algún sentido. Actualmente los investigadores pretenden hacer de la computadora una herramienta eficiente, sin embargo es necesario dotarla con un aprendizaje de la estructura de textos lo que implica el estudio de ciencias del lenguaje y ciencias de razonamiento [13,14].

El análisis documental está constituido por el conjunto de operaciones necesarias para seleccionar las ideas informativamente relevantes de un documento, a fin de expresar su contenido sin ambigüedades, para recuperar la información en él contenida. Esta representación puede ser utilizada para identificar el documento, para procurar los puntos de acceso en la búsqueda de documentos, para indicar su contenido o para servir de sustituto del documento [13, 14].

Por generación automática de resúmenes de texto se entiende que es el proceso por el cual se identifica información sustancial, proveniente de una fuente o varias para producir una versión abreviada. Un resumen debe ser claro, ordenado, expresa lo esencial y refleja los elementos de mayor relevancia sin aportar ideas nuevas.

Existen diversos tipos de resúmenes y uno de ellos es el extracto, el cual se forma a base de frases "extraídas" del propio texto [19,21]. La efectividad de las técnicas de resumen depende del siguiente fundamento: el resumen más apegado al documento original será aquel que sea desarrollado por el propio autor [21]. La extracción de información consiste en la creación de una representación estructurada, a partir de información seleccionada de un conjunto de textos. Actualmente los usuarios de

Internet desean obtener información lo más rápido posible, no desperdiciar su tiempo teniendo que revisar una larga lista de links, que probablemente no sean de su interés. Por ello uno de los objetivos de este trabajo se centra en exponer una metodología [13, 14] para generar extractos de documentos. Para conformar el extracto se siguió la siguiente metodología:

- Conversión de minúsculas: El documento a analizar tiene que ser sometido a un "pre-procesamiento", éste consiste en omitir la combinación de letras mayúsculas y minúsculas y, utilizar solamente letras minúsculas. De esta forma se facilitará el posterior tratamiento del documento y se eliminarán problemas de sensibilidad [6].
- Identificación de tokens y unidad de extracción: Los signos de puntuación operan como delimitadores del lenguaje, y pueden indicar una pausa (la coma o el punto y coma), expresan inicio de una nueva idea (el punto), una aclaración (los paréntesis), etc. En un ambiente de programación los tokens realizan la misma función que los signos de puntuación, permiten dividir una cadena de caracteres en una serie de elementos delimitados por unos determinados caracteres. Este trabajo de investigación emplea como delimitador el punto (.), ya que de esta forma se segmentará el documento en oraciones o frases completas, las cuales fungirán a partir de este momento como la unidad de extracción, debido a que una oración comprende la expresión de una nueva idea de forma coherente y evita problemas de ambigüedad.
- Heurísticas para generación de extractos [21]:
 - Heurística de Posición: Esta heurística consiste básicamente en dar mayor valoración a las primeras frases u oraciones de un texto. En dominios periodísticos, el título y las primeras frases de un texto dan una idea aproximada al lector del contenido del texto que se va a leer a continuación. Por esta razón se seleccionan las N primeras frases del documento que se está resumiendo. En este caso, el sistema opera con el valor típico de N el cual se maneja como cinco, es decir, el extracto contendrá como máximo cinco oraciones las cuales se consideran suficientes para representar el extracto de un documento.
 - Heurística de Palabra Clave: Esta heurística consiste en extraer las frases donde se encuentra la palabra clave, la cual representa una C consulta que el usuario realiza al sistema. Es así como se seleccionan únicamente aquellas oraciones que podrían ser de interés al lector.

Para enriquecer el contenido del extracto se recurrió a utilizar ambas heurísticas de la siguiente forma: *Extraer las 5 primeras frases del documento que contengan la palabra clave*. El algoritmo [13] que ilustra lo anteriormente descrito se muestra a continuación:

- Paso 1. Abrir documento para su lectura.
- Paso 2. Conversión a minúsculas de todo el documento.
- Paso 3. Segmentar el documento en oraciones completas delimitadas por (.)
- Paso 4. Para cada oración identificar si se encuentra la palabra de consulta
 - Si palabra consulta está contenida en oración
 - Incrementar contador de frase de extracto(N)
 - Almacenar tupla en la estructura hasta que $N \leq 5$
 - Si no repetir paso 4.
- Paso 5. Fin

El hilo del objeto activo selecciona alternadamente un método en la cola pendiente de peticiones y lo ejecuta. Por el lado del subsistema, este envía un llamado al OA, este es representado por un *proxy*, el cual tiene como principal responsabilidad generar objetos futuros para representar valores futuros, transformar llamados en objetos Request y ejecutar su copia profunda de objetos pasivos, pasados como parámetros [2].

ProActive proporciona una forma de mover un OA desde cualquier máquina virtual de Java a otra, dicha características se realiza a través de la primitiva `migrateTo(...)`[2, 28]. (Ver Tabla 1.)

Table 1. Primitivas de Migración en ProActive.

	Migración hacia:
<code>migrateTo(URL)</code>	Un Nodo Virtual identificado por un URL
<code>migrateTo(Object)</code>	La ubicación de algún OA

ProActive provee tres mecanismos para mantener comunicación con objetos móviles, el primero refiere a ubicar un servidor el cual guardará el rastro de los objetos móviles en el sistema. El segundo emplea una técnica completamente descentralizada conocida como “forwarders”; cuando se abandona un sitio, un OA deja un objeto especial que llamó un forwarder el cual apunta a su nueva ubicación. Hasta que se recibe un mensaje, este es pasado por el forwarder al objeto (otro forwarder). El tercero es un esquema original basado en los dos mecanismos anteriores y provee tolerancia a fallos [2].

Un OA será desplegado en varios ambientes heterogéneos donde las políticas de seguridad podrían diferir de un lugar a otro, al igual que el cómputo y las comunicaciones varían de host a host. Como tal las ubicaciones de OA efectivos no deben estar atadas al código fuente [2].

El primer principio es eliminar totalmente del código fuente los siguientes elementos [2, 28]:

- Nombres de máquinas
- Creación de protocolos
- Registro y búsqueda de protocolos

El objetivo de esto es desplegar cualquier aplicación en cualquier lugar sin cambiar el código fuente. Se pueden utilizar a su vez diversos protocolos como rsh, ssh, Globus y LSF para la creación de MVJ's necesarias para la aplicación.

Para alcanzar su objetivo, el modelo de programación cuenta con una noción específica de Nodos Virtuales (VN's) [2, 28]:

- Un VN es identificado como un nombre (un simple String),
- Un VN es usado en un programa fuente,
- Un VN es definido y configurado en un archivo descriptor (XML)
- Un VN, después de su activación, es mapeado a uno o a un conjunto actual de Nodos ProActive.

Obviamente, entidades distribuidas (OA), son creados en nodos, no en nodos virtuales. Los VN's son mucho más ricos en abstracción ya que proveen mecanismos tales como un mapeo cíclico, además de la capacidad de describir y disparar los mapeos a un VN que genera la asignación de varias MVJ's.

Por otra parte, un nodo es un objeto que vive en una MVJ, organizando al OA, existe una correspondencia entre los VN's y los nodos: la función creada por el despliegue: el mapeo, el cual puede ser especificado en un descriptor XML. Las operaciones que pueden ser configuradas en un descriptor son las siguientes [2,13,28]:

- El mapeo de VN's a nodos y a MVJ's
- Crear y adquirir MVJ's
- Registrar y buscar VN's

5 Estrategias de lanzamiento de agentes

Los agentes móviles además llevan consigo una lista que determinará los saltos o host por visitar, a ésta se le conoce normalmente como Itinerario del agente [3, 5], además controla casos especiales tales como, qué sucede si un destino no existe, y siempre sabe a donde irá en el siguiente traslado. De igual forma permite el reuso de itinerarios ya que son guardados como agendas. La migración [13], por tanto es la esencia de los agentes móviles, y en este trabajo el enlace para lograr que un agente viaje de un host a otro se realiza por medio de un identificador, en este caso es una dirección IP. Sin embargo el problema de migración es un problema NP-duro en el que se pretende optimizar las rutas de migración de agentes y minimizar los costos totales de comunicación [5, 11, 24].

El sistema es modelado usando agentes del tipo BDI (Beliefs, Desires, Intentions), es decir, Creencias, Deseos e Intenciones [5, 7, 10, 11]. Estas son abstraídas como clases que comprenderán la estructura del agente. Las creencias constituyen el conocimiento del agente sobre sí mismo y su ambiente, en este caso Internet. Los deseos son los objetivos que un agente tiene que cumplir, y para ello se vale de planes que materializan esos deseos y la convierten en una intención, la cual consistirá en generar extractos de documentos.

Para que un agente móvil pueda transitar por una red, es necesario que la plataforma esté apoyada sobre una infraestructura de comunicación determinada, de tal forma que sea dicha infraestructura la que realice la transferencia del agente de una máquina a otra. Las infraestructuras más habituales que utilizan los sistemas de agentes móviles son: TCP/IP o plataformas de más alto nivel como RMI (Remote Method Invocation).

ProActive actualmente utiliza la librería estándar RMI como capa de transporte portable, además trabaja sobre RMI para registrar y encontrar nodos. Un OA siempre es agregado a un nodo el cual representa una entidad lógica en una MVJ. Cuando se crea el OA se le tiene que asociar una URL o una referencia al nodo y este tiene que existir al momento de crear el OA, el cual es lanzado en una MVJ local o remota. [13,16,28]

Para que pueda ser accedido desde una MVJ remota, un nodo automáticamente registra con un RMI registry en la máquina local, la clase que realiza este procedimiento es NodeFactory en el método getNode [28].

Teniendo en cuenta las consideraciones en cómo es que los OA migran, en este trabajo se pretende implementar dos estrategias de generación de OA para extracción de texto en diferentes nodos, a fin de comparar el desempeño respecto al tiempo que son obtenidos los resúmenes.

Como se mencionó anteriormente se diseñó e implementó una clase que busca todos los documentos html almacenados en un host, una vez que los localiza, son abiertos para su lectura y se genera el resumen correspondiente de cada documento, este comportamiento es definido en el método runActivity de la interfaz RunActive, en la cual se especifica la actividad del OA [13, 28].

```
public void runActivity(Body body) {
    Service service=new Service(body);
    while(body.isActive())
    {
        archihtml.buscando();
        service.blockingServeOldest();
    }
}
```

5.1 Primera estrategia de lanzamiento

Se tiene como nodo raíz un host, el cual albergará una Base Datos que guarda extractos de cada documento, a su vez este mismo host fungirá como servidor lanzará diversos OA's, cada uno de ellos representa una dirección IP dentro de la local o bien algún nodo del clúster, es decir, nodos hijos. Posteriormente cada OA creado revisará las diferentes páginas. Cada vez que un OA clonado termina revisar la página, actualiza su información en un arreglo de clase, donde el almacena la dirección URL y el resumen extraído. Una vez que todos los objetos clonados han mandado la información obtenida a su nodo padre o servidor; este actualiza la Base Datos (Ver Figura 1.) [13].

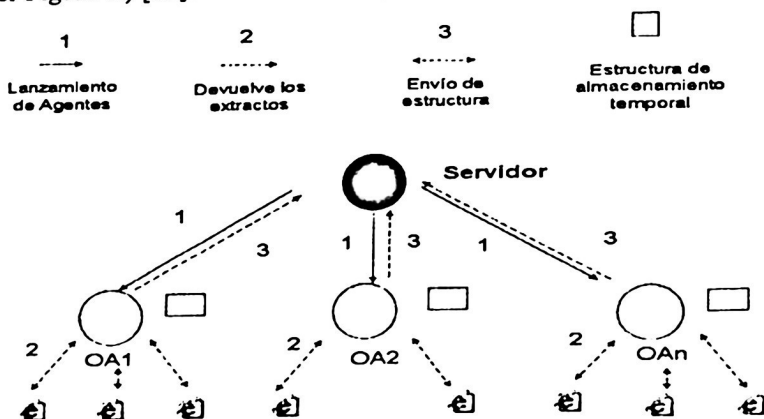


Fig. 1. Primera Estrategia de Lanzamiento. El número indica el orden en que se desarrolla actividad de los agentes.

En la clase Agente se definió el método mover, y como parte de su implementación se hizo uso de la primitiva migrateTo(String Node URL).

```
public void mover(String nodeURL) throws Exception{
    System.out.println(" Voy a migrar");
    ProActive.migrateTo(nodeURL); }
```

En este caso mediante el método mover(...), migra el OA cuyo hilo activo es llamado por este método a el nodo caracterizado por la URL proporcionada. Este método debe ser llamado desde un OA usando su hilo activo como el hilo actual a ser usado para localizar el llamado del OA al método. Es importante remarcar que el parámetro de la URL corresponde a una URL de un nodo existente.

Para especificar el itinerario del agente ProActive, a través de un archivo descriptor XML se detallan las características para definir nodos virtuales, es decir, se establece el número de nodos virtuales y el protocolo de comunicación entre los host [13].

```
<processDefinition id="rloginProcess">
  <rloginProcess
    class="org.objectweb.proactive.core.process.lsf.RLoginP
    rocess" hostname="192.168.0.13">
    <processReference refid="localJVM" />
  </rloginProcess>
</processDefinition>
<processDefinition id="rloginProcess">
  <rloginProcess
    class="org.objectweb.proactive.core.process.lsf.RLoginP
    rocess" hostname="192.168.0.14">
    processReference refid="localJVM" />
  </rloginProcess>
</processDefinition>
...
```

5.2 Segunda estrategia de lanzamiento

Se tiene la misma estructura de árbol mostrada anteriormente, con las siguientes excepciones; los objetos no serán lanzados de forma paralela, sino de uno en uno, una vez que se ha extraído la información de cada página HTML, regresará al servidor con su estructura para actualizar la Base de datos, para esto no es necesario contar con un itinerario, es suficiente con especificar la dirección IP del destino al que se desea migrar [13].

```
org.objectweb.proactive.examples.Migrar.MigrarAgente
//remotehost/node
```

6 Resultados

En lo que concierne a ProActive fueron identificadas algunas restricciones que limitan su desempeño, las cuales se citan a continuación:

- Es necesario utilizar la misma versión de Java en cada uno de los host.
- Es necesario instalar la misma versión de ProActive en cada uno de los host.
- Es necesario que las aplicaciones desarrolladas en ProActive sean ejecutadas ambientes donde se tengan los permisos correspondientes, es decir, acceso determinada LAN, Grid o clúster.
- Es necesario para el lanzamiento de agentes que los host posean un protocolo establecido, ya sea rsh, rlogin, ssh, entre otros.

Sin embargo sus fortalezas radican en la facilidad para especificar protocolos comunicación sin la necesidad de realizar cambios al código fuente. Son una gran cantidad de protocolos los que se pueden emplear y actualmente conforme aparecen nuevas versiones se anexan más protocolos, uno de ellos es SSH y la ventaja proporcionar este tipo de servicio se ve reflejado en cuestiones de seguridad, problema grave de los agentes móviles, pues mediante este protocolo se pueden enviar datos en redes donde no se tiene certeza que sean del todo seguras.

Otra de las ventajas de ProActive son sus requerimientos mínimos, comparados con las bondades obtenidas con su uso, además funciona en sistemas operativos Windows y Unix. Debido a que ProActive posee características multiplataforma pruebas fueron realizadas en el sistema operativo Windows XP y Suse.

Hasta el momento se han implementado las dos estrategias de migración propuestas en este trabajo, las cuales han sido probadas sobre un cluster de 16 nodos donde cada nodo cuenta con 2 procesadores. En estos momentos sólo se están recuperando los resúmenes de las páginas html que se encuentran en el nodo servidor. Uno de los problemas a los que nos hemos enfrentado es que muchos hosts identifican a los agentes móviles como virus por lo cual la segunda estrategia aún se encuentra fase de prueba hasta lograr que los agentes tengan permisos de acceso.

En lo que respecta a la obtención de los extractos se ha presentado una herramienta para obtener las oraciones representativas de un texto, las cuales constituyen lo que llamado un extracto de texto. Las pruebas realizadas hasta el momento han sido realizadas con éxito. Se han revisado todas las páginas html que se encuentran en nodo local obteniéndose resúmenes usando pocos recursos lingüísticos y que brindan una representación cercana a los documentos originales.

7 Conclusiones y recomendaciones

Actualmente la importancia y la necesidad de implantar herramientas en sistemas distribuidos se ha acrecentado considerablemente, esto se ve reflejado en las diversas tecnologías desarrolladas para mantener comunicados equipos remotos que ejecuten tareas y tengan acceso a la información.

Una de las respuestas a este problema es ProActive, que es un API en constante crecimiento y en este momento se encuentra en su versión 2.1 por lo cual cada vez son más las habilidades que proporciona.

A pesar de que ProActive no ofrece herramientas para desarrollar aplicaciones con agentes móviles, con un manejo adecuado de los OA's se pueden crear estas entidades. Es importante considerar que esta API necesita extensiones de vital relevancia para su crecimiento, tales como una infraestructura genérica para agentes móviles, que agregue aspectos de seguridad y responda a situaciones que pudieran causar algún fallo, como por ejemplo si un agente no pudiese entrar a determinado host cómo se tendría que responder.

A su vez el hecho de ligar a los agentes móviles con el área de recuperación de información enriquece el trabajo sobremanera, de tal forma que la tarea de generar extractos de documentos se convierte en uno de los objetivos primordiales del agente (a parte de su migración). El algoritmo presentado sólo muestra las primeras 5 ocurrencias de la palabra de consulta contenida en algún documento, es decir, no analiza el valor informativo que pudiesen aportar otras oraciones. Por el momento no son revisados los contenidos, sólo se obtienen los resúmenes mediante los cuales el usuario decide consultar el documento original.

Una recomendación aunada a lo anteriormente expuesto es el diseño de un agente que revise la información obtenida y decida cual es la que se acerca más a lo que el usuario necesita.

Referencias

1. Arasu, A., Junghoo C., Garcia-Molina H., Paepcke A., Raghavan S.: Searching the Web, ACM Transactions on Internet Technology, Vol. 1, No. 1, August 2001, 2-43.
2. Baduel, L., Baude, F., Caromel, D., Contes, A., Huet, F., Morel, M., Quilici, R. ProActive: Programming, Composin, Deploying for the Grid, OASIS - Joint Project CNRS / INRIA / University of Nice Sophia - Antipolis, 2004, 12-38.
3. Baek, J.W., Kim, G.T., Yeom, H.Y.: Timed Mobile Agent Planning for Distributed Information Retrieval, Proceedings of the Fifth International Conference on Autonomous Agents, ACM Press, 2001, 120-121.
4. Binder, W., Roth, V.: Secure mobile agent systems using Java: where are we heading? Proceedings of the 2002 ACM symposium on Applied computing, ACM Press, 2002, 115-119.
5. Busetta, P., Ramamohanarao, K.: An architecture for Mobile BDI Agents, Proceedings of the 1998 ACM Symposium on Applied Computing, ACM Press, 1998. 445-452.
6. Gilbert, D., Aparicio, M., Atkinson, B., Brady, S., Ciccarino, J., Grosz, B., O'Connor, P., Osijek, D., Pritko, S., Spagna, R., Wilson, L.: IBM Intelligent Agents Strategy, IBM Corporation, 1995.
7. Hagimont, D., Ismail, L.: A Performance Evaluation of the Mobile Agent Paradigm, Proceedings of the 14th ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages and Applications, ACM Press, 1999. 306-311.
8. Horvat, D., Cvetković, D., Milutinović, V., Kočović, P., Kovačević, V.: Mobile Agents and Java Mobile Agents Toolkits, Proceedings of the 33rd Hawaii International Conference on System Sciences, USA, Enero 2000, 1-10.
9. Kobayashi, M., Takeda, K.: Information Retrieval on the Web, ACM Computing Surveys, Vol. 32, No. 2, June 2000. 144-173.

10. Komiya, T., Ohsida, H., Takizawa, M.: Mobile Agent Model for Distributed Systems, Proceedings of 22nd International Conference on Distributed Computing Systems Workshops, 2002. 131-136.
11. Kotz, D., Gray, R., Rus, D.: Transportable Agents Support Worldwide Applications, Proceedings of the 7th Workshop on ACM SIGOPS European Workshop: Systems Support for Worldwide Applications, ACM Press, 1996. 41-48.
12. Kotz, D., Gray, R.S.: Mobile Agents and the Future of the Internet, Department of Computer Science / Thayer School of Engineering Dartmouth College, In ACM Operating Systems Review 33(3), August 1999. 7-13.
13. Leal, L.: Tesis de Licenciatura: "Agentes Móviles para Generar Extractos de Documentos" Universidad Popular Autónoma del Estado de Puebla, Escuela de Ingeniería en Computación, Febrero 2005. 38-66, 73, 79-83, 90-107.
14. Leal, L., Vilariño, D., López, F.: Diseño de Agentes Móviles para Generar Extractos de Documentos en ProActive, Memorias del 2º Congreso Nacional de Ciencias de la Computación, Noviembre 2004, 287-292, ISBN 968 863 798 X.
15. Lu, J.: Some Research on Componentware Frameworks Base on Mobile Agent Technology, ACM SIGSOFT Software Engineering Notes Vol. 29 No. 2, March 2004, pp. 8-8.
16. Márquez, M.: Tesis de Maestría: "Infraestructura Genérica para Agentes Móviles para Aplicaciones Distribuidas bajo ProActive", BUAP Facultad de Ciencias de la Computación, Mayo 2004. 46-58.
17. Nolan, B.: Java and Information Retrieval from the Internet, Proceedings of the 2nd international Conference on Principles and Practice of Programming in Java, Computer Science Press, Inc., 2003. 133-135.
18. Pozo, S., Gasca, R.M., Gómez López, M.T.: Secure Tunnels for Mobile-Agent Systems, Proceedings of the Agent Technology Applications, 5th. Iberoamerican Workshop on Multi-Agent Systems, Iberagents 2004, Noviembre 2004. 12-25.
19. Proux, D., Rechenmann, F., Julliard, L.: A Pragmatic Information Extraction System, Proceedings of Seventh International Symposium on String Processing and Information Retrieval, 2000. 236-241.
20. Russel, S., Norving, P.: Artificial Intelligence A modern Approach, Second Edition, Ed. Prentice Hall Series in Artificial Intelligence, 2003, pp. 1-30, 42-57, 840-850.
21. Salazar, H.: Tesis de Maestría: "Obtención de Extractos de Textos con base en un Corpus", Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Mayo 2004. 1-27.
22. Shoeman, M., Cloete, E.: Architectural Components of Mobile Agent Systems, Proceedings of the 2003 Annual research conference of the South African Institute of Computer Scientists and Information Technologists on Enablement through Technology, South African Institute for Computer Scientists and Information Technologists, 2003. 48-58.
23. Weiss, G.: Multiagent Systems A Modern Approach to Distributed Artificial Intelligence, Massachusetts Institute of Technology, 1999, ISBN 0-262-23203. 1-77.
24. Xu, B., Lian, W. Giang, G.: Migration of Enterprise JavaBeans with ProActive, ACM SIGPLAN Notices, Vol. 38, No. 8, August 2003. 22-28.
25. Yanxiang, H., Yifeng C.: A GA- Based Solution to the Migration Problem of Mobile Agents in Distributed Information Retrieval Systems, Proceedings. 23rd International Conference on Distributed Computing Systems Workshops, Mayo 2003. 466-471.
26. <http://www.fipa.org>
27. <http://www.omg.org>
28. <http://www.sop-inria.fr/oasis/ProActive>

Una nueva ponderación para el modelo de espacio vectorial de recuperación de información

Rubí J. Cabrera, David Pinto, Darnes Vilariño, Héctor Jiménez-Salazar

Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla
14 sur y Av. San Claudio Universitaria, Edif. 135
Puebla, Pue., México, C.P. 72570
Tel. (+52-222)2295500 Ext. 7227, Fax (+52-222)2295672
rubicabrera@hotmail.com, {dpinto, darnes, hjimenez}@cs.buap.mx

Abstract. El modelo planteado busca representar los documentos a través de los términos más importantes del *corpus*, usando un esquema de pesos en cierta manera similar al modelo de espacio vectorial (MEV). Se utilizan dos componentes para el peso de un término en un documento; el primer componente se calcula mediante la distancia inversa del término al punto de transición de su documento, mientras que el segundo componente se basa en la distancia al punto de transición global (considerando todo el *corpus*). Los resultados obtenidos muestran valores de precisión por niveles comparables con el peso clásico *tf-idf*, sin embargo, la dimensionalidad de los términos en nuestro modelo es de tan solo el 3% de la MEV clásico.

Palabras clave: Ponderación de términos, Sistemas de Recuperación de Información.

1 Introducción

1.1 Sistemas de recuperación de información

La creación de los buscadores de información en Internet sigue impulsando la generación de nuevas técnicas para la representación de información y por consiguiente la recuperación de la misma. La forma mas común de encontrar información en Internet es utilizando los llamados "motores" de búsqueda o buscadores. Algunos de los mas populares son: Google, Yahoo, Altavista, Excite, InfoSeek, Web Crawler, entre otros. Existen buscadores en casi todos los idiomas del mundo y algunos de ellos tienen ciertas "especialidades". Los mismos han sufrido grandes mejoras con las diversas investigaciones acerca de la evolución de la información; un caso particular es el de Google el cual realizó mejoras en sus técnicas de indexación automática para lograr y brindar un buen funcionamiento [6][7]. Ciertamente, resulta necesaria la creación de nuevas técnicas y mejoras a las ya existentes para proporcionar información de una forma eficaz ante una búsqueda dada, sobre todo en el lenguaje Español.

Los Sistemas de recuperación de información (SRI) consisten básicamente en un conjunto de procesos interrelacionados que permiten obtener información de interés a partir de una determinada consulta [3]. Un SRI soporta una serie de operaciones sobre la colección de documentos almacenados, como son: introducción de nuevos documentos, reindización de los documentos almacenados y eliminación de los mismos. También debe contar con un método de localización de documentos, para presentarlos según la consulta del usuario. Los SRI implementan estas operaciones en modelos diversos, lo que provoca una amplia variedad en lo relacionado con la naturaleza de los mismos; regularmente es posible encontrar variaciones con respecto a los métodos de búsqueda y técnicas de representación [2].

En este artículo se presenta un nuevo modelo de ponderación de términos de los documentos, basado en la técnica llamada Punto de Transición [9] (que posteriormente abordaremos), así mismo, se realiza una comparación con la ponderación clásica *tf-idf* [2] y se obtiene una evaluación. En la primera sección se muestra la problemática que implica la creación de nuevas técnicas para la representación de documentos; así también, se hace mención de los mecanismos de representación existentes. En la segunda sección se describe el modelo de ponderación propuesto y se define la técnica en la que se basa dicho modelo; se muestran también las fórmulas para el cálculo de pesos que definen al nuevo modelo de ponderación y la definición de la ponderación de la consulta. En la tercera sección se describen los resultados experimentales usando un *corpus* de prueba (TREC {5 [8]}); también se muestra la evaluación del nuevo modelo propuesto, comparándolo con el modelo de espacio vectorial clásico. Por último se realizan una serie de conclusiones en base a los resultados obtenidos y se discuten algunos trabajos a futuro basados en el modelo propuesto.

1.2 Mecanismos de representación

En [6], se refiere a la proliferación considerable en estos últimos años de herramientas para buscar información en la Web; se estima que en la actualidad existen más de 2000 motores de búsqueda diferentes en la Web, mientras que en 1995 había tan solo una docena. Cada uno de ellos tiene sus propias características, utilidades e interfaces de usuario.

Con respecto a la representación de la información almacenada, en [5] se hace referencia a la indización, o indexación, como la operación destinada a representar los resultados del análisis de contenido de un documento o de una parte del mismo, mediante elementos (denominados genéricamente 'términos de indización') de un lenguaje documental o natural, generalmente para facilitar la recuperación.

Un modelo de representación tiene como objetivo satisfacer las necesidades reales y potenciales de información de todos los usuarios, proporcionándoles la información veraz pertinente, justo a tiempo y al menor coste. En particular, contempla una serie de etapas con las que debe cumplir para ser considerado un modelo de representación de RI óptimo [1]:

1. Obtener representación de los documentos. Generalmente los documentos se presentan utilizando un conjunto más o menos grande de términos índice.

2. Identificar la necesidad informativa del usuario. Se trata de obtener representación de esa necesidad, y plasmarla formalmente en una consulta acorde con el sistema de recuperación.
3. Búsqueda de documentos que satisfagan la consulta. Consiste en comparar las representaciones de documentos y la representación de la necesidad informativa para seleccionar los documentos pertinentes.
4. Obtención de resultados y presentación al usuario.
5. Evaluación de los resultados por parte del usuario.

En [7] se hace referencia al gran desarrollo y crecimiento que ha tenido el motor de búsqueda Google como mecanismo de representación de documentos, debido a su extenso uso de la estructura en hipertexto y a su diseño de arrastre e indexación eficiente del Web.

En [6] se describe la paginación de la Web como un método para clasificar las páginas Web objetivamente y mecánicamente con una gran efectividad para el interés humano.

En particular, en la Facultad de Ciencias de la Computación, BUAP, se están desarrollando un conjunto de herramientas destinadas al proceso de RI.

Una técnica particular que se encuentra en investigación en este grupo, es precisamente la del punto de transición. Un trabajo derivado del estudio de esta técnica se puede ver en [4], en donde se presenta un mecanismo para reducir los términos de representación de un documento mediante el PT.

En este artículo se hace uso de la técnica PT para la obtención de un nuevo mecanismo de ponderación comparándolo con el peso clásico del modelo de espacio vectorial propuesto por Salton [2].

2 Modelo propuesto

2.1 Punto de transición

El Punto de Transición (PT) refiere básicamente a un término en el vocabulario del texto que divide al mismo vocabulario en términos de alta y baja frecuencia.

Urbizagástegui [9], por ejemplo, se refiere a este concepto a través de la ley de Zipf [10], y presenta un ejercicio en donde argumenta el hecho de que, existe una vecindad de términos alrededor del punto de transición que describen de manera general el contenido del mismo texto. Este concepto es sumamente importante, ya que dichos términos podrían utilizarse para representar el documento. La fórmula para la obtención del valor de frecuencia del PT se muestra en la ecuación (1).

Donde I_1 es el número de términos que tienen frecuencia 1.

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

En nuestro caso, la representación de cada documento se realiza por medio de un conjunto de términos pertenecientes al vocabulario¹ del mismo documento y que tienen un valor de frecuencia tan cercano al punto de transición como $PT \cdot 0.25$; es decir, se escoge una vecindad del 25% alrededor del PT, la cual resultó la mejor opción después de experimentar con diversos valores de vecindad.

Obtenido el punto de transición con una banda de frecuencia del 25%, deberán seleccionarse términos alrededor de él para conformar el conjunto de palabras que representarían al documento.

2.2 Modelo de ponderación propuesto

Una forma de representar los documentos es por medio del cálculo de pesos, en donde se asigna un valor numérico a cada término del documento. En este caso, se propone establecer un peso sobre los términos alrededor del punto de transición, de acuerdo a la distancia de los mismos hacia el PT. Lo anterior, evaluado por las fórmulas mostradas en (2), (3) y (4).

Así, W_{ij} es el peso que le corresponde al término i en el documento j .

$$W_{ij} = IDPT_{ij} * DPTC_i \quad (2)$$

Donde, $IDPT_{ij}$ es la distancia inversa del término i -ésimo al punto de transición del documento j . Se eleva al cuadrado el denominador para asignar un valor de importancia cuadrático a los términos, en función de su cercanía al PT.

$$IDPT_{ij} = \left| \frac{1}{PT_j - F(t_{ij})^2} \right| \quad (3)$$

$DPTC_i$, por su parte, es la distancia de la frecuencia del término i del vocabulario del corpus al punto de transición evaluado sobre todo el corpus (PT global).

$$DPTC_i = \sqrt{(PT - F(t_i))^2} \quad (4)$$

2.3 Ponderación de la consulta

Los usuarios que consultan a través del sistema de recuperación (SRI) para buscar información, deben traducir su necesidad informativa en una consulta adecuada al SRI. Esto supone utilizar un conjunto de términos que expresen semánticamente su necesidad. En sistemas tradicionales es habitual utilizar un valor de peso asociado a

¹ El vocabulario de un texto es el conjunto de palabras no repetidas del mismo documento.

cada término de la consulta. En este caso se asigna el peso usando el cálculo definido en la sección 2.2, a excepción de que $IDPT_{ij}$ se asume con un valor de 1, debido principalmente a la cantidad y calidad de términos usualmente contenidos en una consulta.

3 Resultados experimentales

3.1 *Corpus* de prueba

TREC (Text Retrieval Conference) constituye uno de los esfuerzos más significativos de investigación experimental en recuperación de información (RI). El patrocinio de elaboración de estas conferencias se encuentra a cargo de la National Institute of Standards and Technology (NIST) y de la Defense Advanced Research Projects Agency (DARPA). Dichas conferencias comenzaron en 1992 (TREC-1) y vienen celebrándose con periodicidad anual hasta la fecha. De manera particular, en 1996 se celebró TREC-5, en cual se utilizó una colección de aproximadamente 250,000 noticias en español. El TREC-5 posee alrededor de 50 consultas supervisadas y las respuestas son indicadas en el mismo *corpus* [8]. La idea es establecer comparaciones fiables entre los distintos sistemas empleados por los investigadores en TREC-5, dado que todos operan con las mismas colecciones y las mismas consultas, y presentan sus resultados en la misma forma; obviamente, utilizan sistemas y técnicas diferentes. En base a lo anterior se creó un *corpus* basado en el TREC-5 para la realización de las pruebas, comprendido de 884 noticias correspondientes al Diario el Norte de Guadalajara, de las cuales el 33% consiste de las noticias relevantes de las consultas 26 y 28; el 67% restante consiste de noticias no relevantes para estas mismas consultas. Los resultados obtenidos se muestran en la siguiente sección.

3.2 Consultas utilizadas

El SRI fue evaluado con las siguientes dos consultas tomadas del TREC-5.

1. Indicaciones de las relaciones económicas y comerciales de México con los países europeos. (Consulta 26)
2. Indicaciones de las relaciones económicas y comerciales de México con los países asiáticos, por ejemplo Japón, China y Corea. (Consulta 28)

3.3 Evaluación con el MEV

Se utiliza una gráfica de precisión por niveles estándar para comparar el nuevo modelo de ponderación de términos en documentos contra el modelo de espacio vectorial propuesto por Salton [2].

Las gráficas muestran que ambos sistemas obtienen resultados comparables, sin embargo, es importante remarcar que el modelo de ponderación propuesto usa únicamente el 3% de términos que usa el modelo de espacio vectorial clásico.

Esto implica incluso que los tiempos de cálculo de representación de documentos se reducen drásticamente.

En la figura 1, se presenta la evaluación sobre la consulta 26 del TREC-5, mientras que en la figura 2 se evalúa sobre la consulta 28.

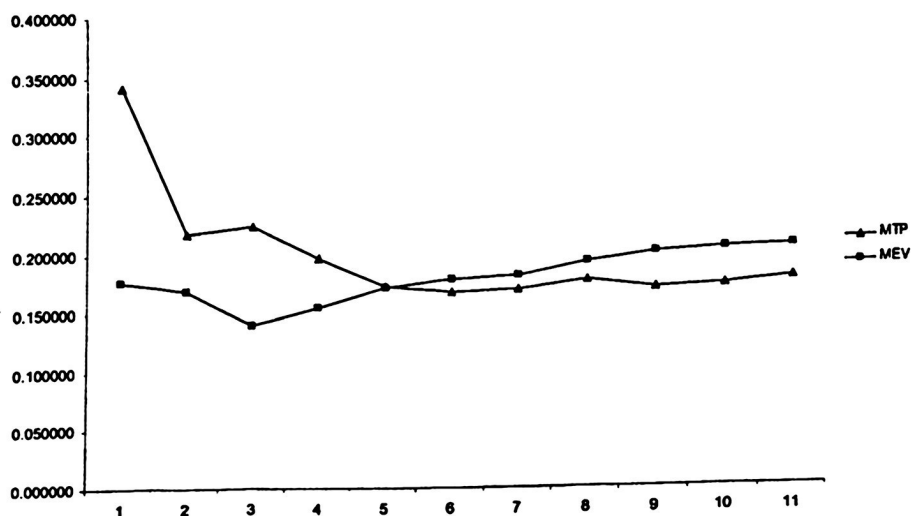


Fig. 1. Consulta 26.

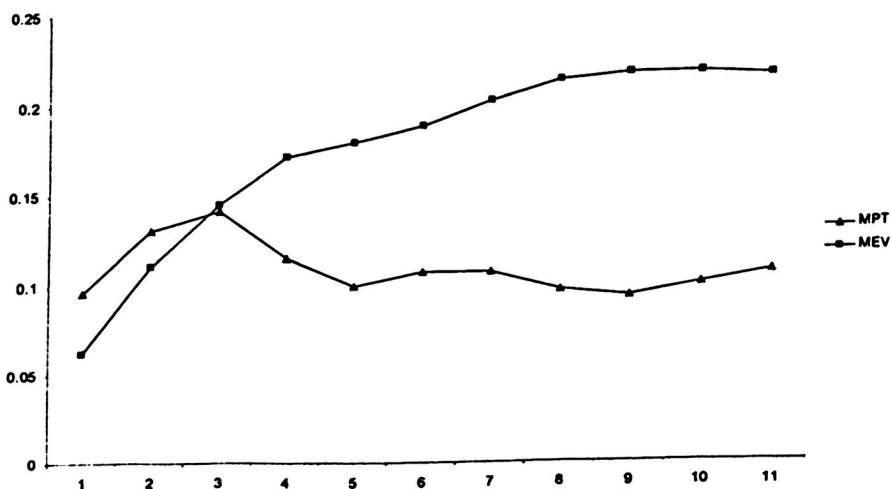


Fig. 2. Consulta 28.

4 Conclusiones

Se ha presentado un nuevo modelo de ponderación de términos de documentos que obtiene una reducción de la dimensión del espacio vectorial bastante amplia.

El efecto de reducción de términos proviene de la utilización de una técnica llamada punto de transición. El modelo usa una fórmula que asigna pesos a los términos utilizando dos componentes para medir la importancia de un término en un *corpus*. El primer componente mide la importancia de un término en un documento, obteniendo la distancia inversa de la frecuencia del término al PT del mismo documento. Por otro lado, el segundo componente mide la importancia del término en todo el *corpus*, y esto lo hace a través de la distancia hacia el PT global (i.e., considerando todo el *corpus*). Los resultados obtenidos muestran que el nuevo modelo de ponderación es comparable con el modelo clásico *tf-idf* en cuanto a la precisión por niveles estándar, sin embargo, el nuevo modelo posee una cardinalidad sumamente menor que el MEV clásico. En este caso, el MEV clásico tiene vectores de representación con 30,202 términos, mientras que el nuevo modelo únicamente usa 1,185.

Es necesario hacer mas pruebas que revelen los alcances de esta propuesta de ponderación. Por ejemplo, la determinación del umbral alrededor del punto de transición, considerar pruebas con todos los tópicos del TREC-5 y ajustar la fórmula de pesos para ganar eficiencia en el cálculo de índices.

Referencias

1. Angel F., Rodríguez Zazo, Figuerola G., Alonso J.L. and Gómez R., *Recuperación de Información utilizando el Modelo Vectorial*, Departamento de informática y automática, Universidad de Salamanca, 2002, Mayo.
2. G. Salton, *Automatic Text Processing*, Addison-Wesley,(1989).
3. Jiménez H. and Pinto D., *Notas de Academia Recuperación de Información*, octubre,(2003).
4. Moyotl E., Reyes B. and Jiménez H., *Reducción de términos índice usando el Punto de Transición*, (2003).
5. Normas Fundamentales, Norma UNE 50-113-92 *Documentación e información*, AENOR, 23-73, vol. 2,(1997).
6. *The Pagerank citation Ranking Bringing Order to Web*, January,(1998).
7. Sergey Brin and Lawrence Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, Computer Science Department, Stanford University, Stanford, CA 94305, USA. [http://trec.nist.gov/Text Retrieval Conference \(TREC\)_ultima revision 13/sep/04](http://trec.nist.gov/Text Retrieval Conference (TREC)_ultima revision 13/sep/04).
8. Urbizagategui A. R., *Las implicaciones de la ley de Zipf en la indización automática*, Universidad de California Riverside,(1999).
10. Zipf George K., *Humand Behavior and the Principle of Least-Effort*, Addison-Wesley, Cambridge MA,(1949).

El párrafo virtual en la generación de extractos

Claudia Bueno-Tecpanecat¹, David Pinto², Héctor Jiménez-Salazar³

Facultad de Ciencias de la Computación,
Benemérita Universidad Autónoma de Puebla,
14 sur y Av. San Claudio, Ciudad Universitaria, Edif. 135
Puebla, Pue., México, C.P. 72570

Tel. (+52-222)2295500 Ext. 7212, Fax (+52-222)2295672

¹claudiabt@mail.cs.buap.mx, ²dpinto, ³hjimenez}@cs.buap.mx

Resumen. En este artículo se presenta un método para la generación automática de extractos usando un concepto novedoso al que hemos denominado *párrafo virtual*; éste está compuesto de términos considerados representativos del contenido semántico del texto del cual se pretende obtener el extracto. Se usa como *corpus* un subconjunto de noticias provenientes del TREC-5 y se evalúan los resultados mediante un sistema de recuperación de información (SRI) basado en el modelo de espacio vectorial (MEV). La hipótesis de verificación consistió en evaluar un par de consultas supervisadas pertenecientes al mismo TREC-5 usando dos *corpora*: documentos completos y solamente extractos. Los resultados obtenidos son realmente alentadores, ya que posteriormente a la evaluación de las consultas, se observa que incluso en algunos casos, la precisión por niveles estándar es mejor cuando el SRI utiliza el conjunto de extractos como *corpus*.

Palabras clave: Extractos, Punto de Transición, Recuperación de Información.

1 Introducción

El crecimiento desmesurado de Internet ha generado grandes volúmenes de información, difíciles de manipular aún para sistemas de búsqueda de información tan complejos como Google [10].

De esta manera, cuando se realizan consultas en estos sistemas, surge el problema de obtener información, que en su mayoría carece de significado para el usuario. Este fenómeno ha impulsado a investigar y desarrollar aplicaciones en el campo de tecnologías de recuperación y extracción de información que permitan obtener únicamente la información requerida [5].

El empleo del resumen de un documento es atractivo, pues reduce el espacio de almacenamiento y facilita el acceso a la información relevante. En este trabajo, usamos el concepto de extracto [3] para la elaboración de resúmenes automáticos.

El proceso de generación de extracto de un documento consiste en obtener oraciones pertenecientes al mismo documento consideradas como de mayor relevancia [3]. Para nuestro caso, determinamos la importancia de los párrafos de

un documento a través de una función de similitud entre éstos y lo que hemos denominado como párrafo virtual (PV).

El PV está compuesto de un conjunto de términos de alta importancia semántica para el texto en cuestión, los cuales se obtienen a través de la técnica del punto de transición [8].

Para la elaboración de este trabajo se ha tomado como referencia una copilación de trabajos previos de extracción, como es el caso de [4] en donde se presenta una propuesta para generar el extracto de un texto. Este trabajo principalmente se basa en el uso de una técnica denominada "expansión por corpus", que sirve para expandir cada término de un documento en base a su sentido. Las oraciones expandidas son procesadas, después se usa una función de similitud para la obtención de oraciones más significativas que constituyen el extracto del texto. En otro proyecto [7], se utilizan dos técnicas para la obtención del extracto de un texto y posteriormente se realiza una serie de consultas sobre los extractos para evaluar los resultados. La idea es innovadora, sin embargo, la cantidad de documentos utilizados es sumamente pequeña y adicionalmente, la evaluación de los juicios es realizada únicamente por tres jueces. Los resultados presentados en ese trabajo son interesantes, valdría la pena verificar el comportamiento de los algoritmos con un corpus mucho más grande. Una aportación más se encuentra en [6], donde se realiza una comparación de dos métodos que determinan automáticamente el extracto de un texto. El objetivo de ese trabajo es hacer patente la importancia del título de un texto. Se obtienen extractos de un texto usando funciones de similitud entre todas las oraciones (método uno) y usando una función de similitud entre el título del texto y las oraciones restantes (método dos). Los resultados muestran que ambos resultados son semejantes, sin embargo, el método dos se encuentra en un orden de complejidad lineal a diferencia del primer método que se encuentra en uno cuadrático.

Como se puede observar, existen diversos trabajos en generación automática de extractos y con resultados interesantes, que alientan a continuar investigando, con la finalidad de mejorar el desempeño.

Este artículo se inspira en los trabajos presentados en [6], con la finalidad de construir un sistema de complejidad lineal (con respecto al número de párrafos) para la obtención del extracto de un texto. El componente innovador del trabajo radica en la generación de un párrafo virtual que representa el significado de dicho documento. La calidad de los extractos obtenidos se evalúa mediante un SRI basado en el MEV. Dado un conjunto de consultas supervisadas, se asume que los extractos son de buena calidad si la precisión obtenida en el SRI es comparable a la precisión en el mismo SRI, usando como *corpus* al conjunto de documentos completos.

En las siguientes secciones se puntualiza dicha investigación. En la sección 2 se explica la descripción del método propuesto, en la sección 3 se presenta la descripción de los datos (*corpus* y consultas), la sección 4 muestra los resultados experimentales y por último las conclusiones y trabajos futuros son abordados.

2 Descripción del método propuesto

2.1 La técnica del punto de transición (PT)

El PT surge a partir de las observaciones de George Kinsley Zipf, quién formuló la ley de frecuencias de palabras de un texto (Ley de Zipf), donde establece que las palabras con mayor frecuencia absoluta son las palabras cerradas, mientras que las menos frecuentes son aquellas que reflejan el estilo y riqueza del vocabulario y por último las que aparecen en la zona media de la función de distribución de frecuencias son las que representan a los documentos [8]. El PT es la frecuencia de un término del texto que divide en dos a los términos de un vocabulario (en términos de alta y baja frecuencia). Esto significa que los términos más cercanos al PT, tanto de alta y baja frecuencia, pueden ser usados como palabras clave que identifiquen a un documento.

La fórmula usada para el Punto de Transición es la siguiente:

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

donde I_1 representa el número de palabras que tienen frecuencia 1.

Booth [2] derivó la ley de términos de baja frecuencia de la cual proviene la ecuación 1, sin embargo, se presenta un inconveniente con respecto a los documentos que son demasiado pequeños, ya que para este tipo de textos, el valor obtenido para el PT regularmente se encuentra fuera de las frecuencias obtenidas en su vocabulario. A partir de este trabajo se observa que el PT puede ser obtenido por inspección, eligiendo dentro del vocabulario el primer término con la frecuencia más baja que no se repita. A partir de este punto se toma un porcentaje del 25% de términos de alta y 25% de baja frecuencia para obtener un rango de transición. Los experimentos realizados muestran que al tomar una banda de frecuencias desde un 15% hasta un 25% alrededor de PT, se obtienen los mejores resultados [1].

2.2 Método para la generación de extractos

El proceso de generación del extracto de un texto se presenta a continuación:

- Preprocesamiento. Se realiza el preproceso de cada uno de los documentos del *corpus* mediante la eliminación de las palabras cerradas (artículos, preposiciones, etc.), y el particionamiento del documento en párrafos.
- Obtención del vocabulario. Se calcula la frecuencia de ocurrencia de cada término no repetido que posea el documento preprocesado.
- Generación del párrafo virtual. Se aplica la modalidad para documentos pequeños del PT, tomando el término con la frecuencia más baja que no se repite de los términos del vocabulario de cada documento. Así, se obtienen los términos significativos o que se encuentren dentro de una vecindad del 25% alrededor del PT, los cuales constituyen lo que denominamos el párrafo virtual.

- Determinación de párrafos significativos. Se utiliza la función de similitud Jaccard (ver ecuación 2) usada comunmente en el modelo booleano de representación de información [9], para determinar que tanto se parecen dos elementos. En este caso se compara cada párrafo del documento con el párrafo virtual que se obtuvo del PT. Los extractos de cada documento se conforman mediante los tres párrafos más relevantes.

$$sim(D, q) = \frac{\#(D \cap q)}{\#(D \cup q)} \quad (2)$$

3 Descripción de los datos

3.1 Corpus

El TREC es una colección de documentos y de consultas supervisadas cuyo único propósito es apoyar a la investigación dentro de la comunidad del procesamiento del lenguaje natural, proporcionando la infraestructura necesaria para la evaluación de metodologías de la recuperación de información.

Existen diversos TREC que se han llevado a cabo año con año; de esa variedad se eligió el TREC-5. Este *corpus* consta de una recopilación de noticias (aproximadamente de 250 megabytes de información) que se obtuvieron del periódico mexicano “El Norte” de Guadalajara y 300 megabytes del periódico de “Agence France Presse”.

Aunque el TREC-5 consiste de 230,820 documentos, en nuestro caso obtuvimos un subconjunto para las pruebas. Nuestro *corpus* consta de 884 noticias, de las cuales el 33% consiste de las noticias relevantes de las consultas 26 y 28 del mismo TREC-5, y el 67% restante consiste de noticias no relevantes para las mismas consultas. Una descripción más detallada de estas consultas se presenta a continuación.

3.2 Consultas utilizadas

Una consulta es una oración en lenguaje natural que sirve para obtener información de interés. Para este artículo se trabaja con las siguientes dos consultas tomadas del TREC-5.

1. *Indicaciones de las relaciones económicas y comerciales de México con los países europeos.* (Consulta 26)
2. *Indicaciones de las relaciones económicas y comerciales de México con los países asiáticos, por ejemplo Japón, China y Corea.* (Consulta 28)

4 Experimentos

4.1 Evaluación de resultados

Es realmente complicado realizar una evaluación sobre la calidad de extractos obtenidos de un *corpus*. En [5] y [7], por ejemplo, se utilizan jueces humanos que generan su propio extracto, sin embargo, se observa un alto grado de desacuerdo entre ellos.

En este trabajo se evalúa la calidad de los extractos usando un SRI construido sobre dos *corpora* (el primero constituido del *corpus* completo y el segundo de los extractos de este mismo *corpus*).

Este SRI calcula los pesos de representación de los documentos mediante el MEV propuesto por Salton [9] y evalúa la relevancia de los mismos por medio de la fórmula de similitud del coseno del ángulo entre los vectores.

Nuestra hipótesis consiste en obtener gráficas de precisión y evocación similares. Si este es el caso, significará que la calidad del extracto es buena.

Tabla 1. Algunos ejemplos de párrafos virtuales

# de doc.	# de noticia	párrafo virtual
1	0000054	exportaciones estados unidos
2	0000182	presidente carta publicación económica económico inversión fiscal estados economistas unidos ritmo
3	0004750	empresarios europeos europea méxico
4	0008846	rendimiento incremento cartera 1992 emisión
.	.	.
.	.	.
.	.	.
884	0202886	china permitirá familiar dijo solidaridad sistema méxico

Después de haber obtenido los párrafos virtuales, se aplica la similitud de Jaccard dentro de cada documento para generar los extractos (mediante los tres párrafos más relevantes); véase el ejemplo siguiente de tres noticias con sus respectivos extractos.

1. La noticia SP94-0000054:

- extracto 1. Indicó dirigente Camexa que los problemas para exportar Europa son conocimiento parcial los mexicanos las obligaciones para exportar las diferencias idioma transporte.
- extracto 2. Entrevistado durante presentación Feria internacional maquinaria para envase embalaje empresario sealó que Alemania requiere productos textiles.
- extracto 3. industrial mexicano debe voltear sus ojos sólo los países hispanos especialmente Estados Unidos tiene que hacer esfuerzo para posesionarse mercado europeo insistió.

2. La noticia SP94-0000182:

- extracto 1. creemos firmemente que plan política fiscal que contenga controles más rgidos sobre gasto federal una reducción del índice tributario fiscal IRA retiro individual) universalmente disponible incentivos eficientes efectivos para creación impuestos inversión empresaria influiría mucho para comenzar renovación Estados Unidos informó carta del AIV
- extracto 2. una carta Presidente grupo comercial Wall Street recomendó que política fiscal 1993 enfoque reducción del déficit implantación incentivos inversión ahorro promoción creación impuestos inversión empresaria
- extracto 3.sugieren reducción déficit Asociación Industria Valores recomendó consideración del Presidente Clinton sobre una serie medidas fiscales alentar crecimiento económico asegurar que Estados Unidos fortalezca papel como potencia económica líder

3. La noticia SP94-0004750:

- extracto 1.Estados Unidos agregó depende poco más sus exportaciones Europa bloque asiático
- extracto 2.Recordó que bloque estadounidense animal muy distinto bloque europeo donde las relaciones económicas equilibrio poder entre las naciones son más balanceadas las que Estados Unidos tiene que enfrentar Continente Americano
- extracto 3.Unidos tuvo que formar propio bloque para competir pero podrá funcionar pronto como una sola nación porque tiene todavía que recorrer distancias económicas culturales que separan sus socios comerciales reiteró

4.2 Pruebas realizadas

Despues de evaluar las dos consultas supervisadas sobre el SRI, se observó que para el caso en que se utilizan los documentos completos como *corpus* se obtiene una baja precisión y una alta evocación global; este resultado es una consecuencia del alto número de documentos obtenidos (ver tabla 2), ya que de un total de 884 documentos se obtienen alrededor de 700.

Tabla 2. Evaluación sobre el corpus con documentos completos

	Consulta 26	Consulta 28
Documentos relevantes de acuerdo al TREC	154	155
Documentos relevantes de acuerdo al SRI	712	720
Presición global	0.1994	0.19444
Evocación global	0.92207	0.90322

En la tabla 3 se presenta la evaluación sobre el *corpus* que posee solamente los extractos; en este caso, la precisión global es mayor que en el caso anterior, la cantidad de documentos obtenidos por el SRI es mucho menor que en el caso anterior, lo cual supone una mejora en el proceso de filtrado sobre el conjunto de documentos relevantes.

Tabla 3. Evaluación sobre el corpus de extractos

	Consulta 26	Consulta 28
Documentos relevantes de acuerdo al TREC	154	155
Documentos relevantes de acuerdo al SRI	272	282
Precisión global	0.224265	0.223404
Evocación global	0.396104	0.406452

Las figuras 1 y 2 muestran el comportamiento del SRI sobre ambos *corpora*, para las consultas 26 y 28 del TREC-5.

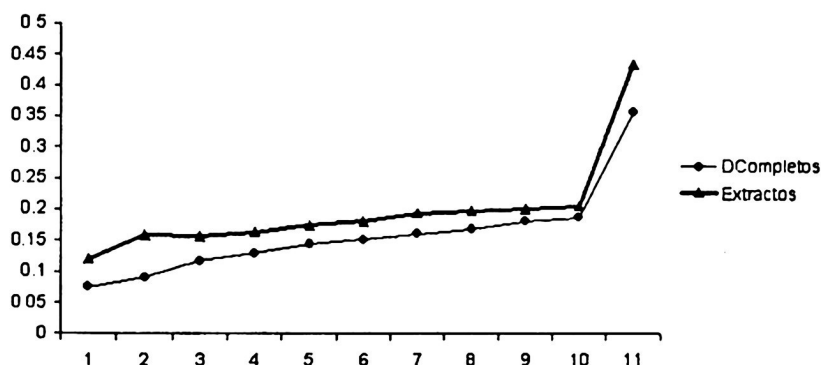


Fig. 1. Gráfica de precisión por niveles estándar para la consulta 26

5 Conclusiones

Se ha presentado un método novedoso para la generación de extractos de documentos. El método hace uso de la técnica del punto de transición para generar una oración virtual que es utilizada para obtener el conjunto de párrafos más representativos de un texto. Los tres párrafos más relevantes se consideran el extracto del texto.

Los resultados obtenidos muestran un comportamiento alentador, ya que a pesar de que se esperaba que el uso de extractos como *corpus* obtuviese una precisión por niveles estándar ligeramente por debajo de la obtenida por los documentos completos, se observó un incremento de la precisión. Esto presume que el método propuesto ha logrado eliminar términos de los textos que efectivamente no tienen un valor significativo de representación para el mismo documento. Se deberán validar estos resultados en todo el TREC-5, a fin de robustecer las conclusiones obtenidas.

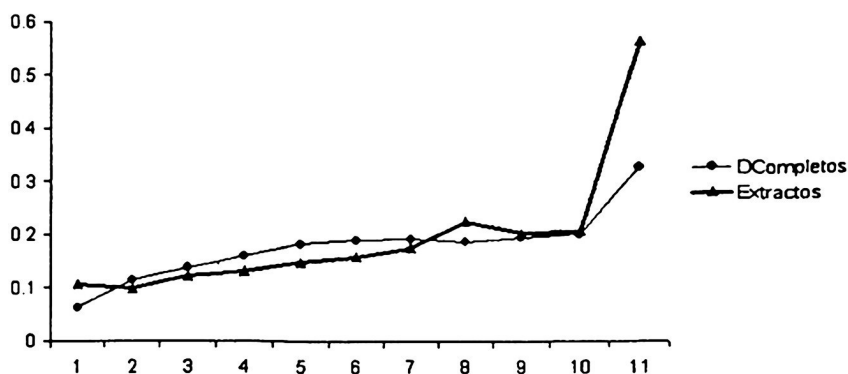


Fig. 2. Gráfica de precisión por niveles estándar para la consulta 28

A pesar de que el uso del SRI fue con la finalidad de medir la calidad de los extractos, se observó que el uso de extractos como mecanismo de representación de documentos puede ser una buena alternativa en la recuperación de información.

Referencias

1. Berenice, Reyes-Aguirre., Edgar, Moyotl-Hernández & Héctor, Jiménez-Salazar.: Reducción de Términos Índice Usando el Punto de Transición, *Facultad de Ciencias de Computación*, BUAP (2003).
2. Booth, Andrew.: A Law of Occurrences for Words of Low Frequency, *Information and control* (1967).
3. Climent-Roca, S.: Sistemas de Resumen Automático de Documentos, *revista digital d'Humanitats*, 3(1) (2001).
4. Héctor, Jiménez-Salazar., Hilario, Salazar-Martínez. & David, Pinto-Avendaño.. Text Extraction A Corpus-Based Approach, *XXX aniversario del programa educativo de computación*, BUAP (2003).
5. Héctor, Jiménez-Salazar, David Eduardo, Pinto Avendaño: Recuperación de Información, *Notas de la Academia* (2003).
6. Hilario, Salazar-Martínez., David, Pinto-Avendaño., Héctor, Jiménez-Salazar.: Comparación de dos métodos que determinan automáticamente el extracto de un texto. *Taller de tecnologías del Lenguaje Humano*, (ENC 2004), ISBN:970-692-170-2 (2004), 267-273.
7. Jiménez-Salazar, H., Pinto-Avendaño, D., Salazar-Martínez, H.: Information Retrieval Based on Text Extraction, *1st. Indian International Conference on Artificial Intelligence*, (IICAI'03), ISBN:0-9727412-0-8(2003).
8. Ruben, Urbizagástegui.: Las posibilidades de la Ley de Zipf en la indización automática, *Reporte de la Universidad de California Riverside* (1999).
9. Salton, Gerard.: Advanced Information-Retrieval Models, *Automatic Text processing*, (1989).
10. Sergey, Brin., Lawrence, Page.: The anatomy of a large-scale hypertextual web search engine. *Computer Science Department, Stanford University*, Stanford, C 94305, USA (1998).

Un lenguaje de apoyo para tareas de preprocesamiento de textos

Erika Hernández¹, David Pinto², Héctor Jiménez-Salazar³, Jesús Lavalle⁴

Facultad de Ciencias de la Computación,
Benemérita Universidad Autónoma de Puebla,
14 sur y Av. San Claudio, Ciudad Universitaria, Edif. 135
Puebla, Pue., México, C.P. 72570

Tel. (+52-222)2295500 Ext. 7212, Fax (+52-222)2295672

¹erikab@mail.cs.buap.mx, {²dpinto, ³hjimenez, ⁴jlavalle}@cs.buap.mx

Resumen. El tratamiento de textos en Internet para su uso posterior en Sistemas de Recuperación de Información (SRI), es de vital importancia. Es un hecho que los resultados de precisión y evocación en los SRI dependen en gran medida de la fase de preprocesamiento. En este proyecto se propone un lenguaje computacional que sirva como herramienta de apoyo en tareas de preprocesamiento de textos. Se ha definido una gramática e implementado el proyecto mediante los lenguajes Java y C (a través de flex y bison). Se han realizado experimentos sobre diversos *corpora*: TREC-5, EuroGOV y otras colecciones compiladas manualmente. Los resultados obtenidos muestran una aceleración y mejora en la etapa de preprocesamiento.

Palabras clave: Procesamiento del Lenguaje Natural, Preprocesamiento, *Corpora*.

1 Introducción

Con el fin de desarrollar sistemas robustos para el Procesamiento de Lenguaje Natural (PLN), es necesario emplear algoritmos que comúnmente requieren que los documentos fuente sean preprocesados antes de utilizarlos en alguna técnica de análisis de información. Adicionalmente, es posible verificar que los conjuntos de información existentes carecen de una estructura y simbología homogénea que permita crear un programa *ad-hoc* para esta tarea. El preprocesamiento universal de *corpora* es una tarea difícil y aun por resolver; de hecho, investigadores en el área de PLN requieren programar métodos de preprocesamiento diferentes, cada vez que necesitan aplicar alguna técnica de análisis de información a nuevos *corpora*.

De manera general, el preprocesamiento puede ser planteado de la manera siguiente: dado un texto D , denotamos con D' al que se obtiene por eliminar las palabras cerradas, símbolos y etiquetas [1].

En algunos trabajos, como el presentado en [2], se contemplan las siguientes operaciones de preproceso: separación de palabras utilizando el carácter de espacio, detección de nombres simples, detección de nombres formados con la inclusión "de/de la" y la eliminación de palabras cerradas. Por otro lado, en [3], se

utiliza un vocabulario formado por las palabras clave que describen a todos los vocablos de un glosario, eliminando de éste, las palabras cerradas y aplicando el algoritmo de Porter adaptado al español para realizar truncamiento.

El tiempo empleado en programar métodos de preprocesamiento para *corpora* heterogéneos crece de manera sustancial, produciendo una inversión de tiempo considerable en la fase de preprocesamiento, el cual podría reducirse para incrementar el periodo de pruebas. Es por ello que el presente trabajo expone un lenguaje con la capacidad de preprocesar una gran variedad de *corpora*, con la finalidad de convertirlos en textos planos preprocesados y listos para tareas específicas de PLN. La gramática brinda una sintaxis casi en Lenguaje Natural(LN), lo que hace de éste un lenguaje de fácil aprendizaje.

El resto del artículo se organiza de la siguiente manera. En la sección 2 se describe el lenguaje desarrollado, exponiendo la gramática como parte fundamental en el diseño del lenguaje. En la sección 3 se presenta la implementación de la gramática. En la sección 4 se muestra un conjunto de pruebas aplicadas sobre diferentes *corpora*. Finalmente las conclusiones del trabajo son presentadas.

2 Diseño de la gramática

2.1 Descripción del lenguaje

En la herramienta diseñada, se han incluido tres archivos (*stopwords.txt*, *symbols.txt* y *labels.txt*), los cuales contienen elementos iniciales que, se espera, faciliten la labor de preprocesamiento de información. Estos contienen palabras cerradas¹, símbolos² y etiquetas de formato³, los cuales cuentan con un tamaño inicial de 20KB, 9KB y 7KB, respectivamente. Se han diseñado diversos mecanismos para administrar dichos archivos. A continuación se puntualiza sobre cada uno de ellos, así como también se hace mención de los servicios soportados por el lenguaje.

Los servicios que el lenguaje proporciona se han clasificado de la siguiente manera: servicios a nivel de primitivas y servicios a nivel de preprocesamiento.

Servicios a nivel de primitivas: Fueron denominados así por el hecho de permitir únicamente el manejo de las operaciones básicas, efectuadas sobre las colecciones de palabras cerradas, símbolos y etiquetas de formato. A continuación se describen las operaciones brindadas por este servicio.

La herramienta proporciona tres archivos o colecciones⁴, permitiendo a los usuarios agregar, eliminar y consultar elementos de estas colecciones.

¹ También conocidas como *stopwords*, son palabras que aparecen con frecuencia entre los documentos y no son buenas para la recuperación de información (artículos, preposiciones, conjunciones, etc).

² Se consideran signos de puntuación, caracteres ASCII y algunos otros símbolos localizados en los textos.

³ Conjunto de marcas que forman parte de la estructura de algunos *corpora* por ejemplo: <title>, < \ title> <p>, < \ p>, <TEXT>, <\ TEXT>, etc.

⁴ *stopwords.txt*, *symbols.txt* y *labels.txt*.

Servicios a nivel de preprocesamiento: Estos tipos de servicios permiten la preparación de *corpora*, mediante la invocación de los servicios primitivos cada vez que sea necesario.

Operaciones sobre corpora: El lenguaje permite la creación de un nuevo *corpus* como un contenedor de archivos. Acepta la eliminación de un *corpus* ya existente en caso que así se requiera. Proporciona un listado de archivos pertenecientes a un determinado *corpus*. Muestra el contenido de un archivo perteneciente a un *corpus*, así como las propiedades⁵ del mismo. Permite la inserción de nuevos archivos a diferentes *corpus*.

Operaciones de preproceso: El lenguaje permite preprocesar un determinado archivo perteneciente a un *corpus*, bajo diferentes niveles de preprocesamiento tales como: eliminación de palabras cerradas, eliminación de símbolos, eliminación de etiquetas de formato, así como cualquier combinación de las operaciones anteriores. Extrae sólo partes de un determinado archivo perteneciente a un *corpus*, mediante la inclusión de cadenas de inicio y fin. Permite la sustitución de una determinada etiqueta por otra, mediante la inclusión de cadenas de inicio y fin.

Para hacer uso de los servicios, a nivel primitivo y a nivel de preproceso, es necesario contar con una gramática que verifique las sentencias introducidas por el usuario para aceptarlas o en su defecto rechazarlas, por lo que en la siguiente sección se introduce la estructura de la misma.

2.2 Estructura de la gramática

El establecimiento de las reglas gramaticales que el lenguaje debe de cumplir es una de las situaciones más complejas para su desarrollo. La forma en que se describe la gramática en este trabajo está dada por la notación BNF (Backus-Naur-Form).

La definición de la gramática libre de contexto (GLC) está dada por:

$$G = (V_n, V_t, S, P)$$

donde

V_n = Un conjunto de símbolos no terminales,

V_t = Un conjunto de símbolos terminales,

S = Símbolo inicial de la gramática,

P = Un conjunto de producciones.

A continuación se presenta la GLC del lenguaje desarrollado.

$S = \langle \text{SENTENCIA} \rangle$

$$V_n = \left\{ \begin{array}{l} \langle \text{SENTENCIA} \rangle, \langle \text{CREATE} \rangle, \langle \text{DROP} \rangle, \langle \text{ADD} \rangle, \langle \text{DELETE} \rangle \\ \langle \text{LIST} \rangle, \langle \text{INSERT} \rangle, \langle \text{DISPLAY} \rangle, \langle \text{PREPARE} \rangle, \langle \text{REMOVE} \rangle \\ \langle \text{REPLACEALL} \rangle, \langle \text{EXTRACT} \rangle, \langle \text{ID} \rangle, \langle \text{USING} \rangle, \langle \text{CICLOS} \rangle \\ \langle \text{LETRA} \rangle, \langle \text{DIGITO} \rangle, \langle \text{NOMARCH} \rangle, \langle \text{STRING} \rangle, \langle \text{SIMBOLO} \rangle \\ \langle \text{DICCIONARIOS} \rangle \end{array} \right\}$$

⁵ Tales como: ruta, tamaño del archivo y fecha de la última modificación.

$$V_t = \left\{ \begin{array}{l} \text{create, corpus, drop, list, display, add, from, delete, insert, put, prepare} \\ \text{and, extract, to, replaceall, for, acutes, a, b, c, d, e, f, g, h, i, j, k, l, m, o, p, q, r, s, t, u, v, w, x, y} \\ \text{z, A, B, C, D, E, F, G, H, I, J, K, L, M, , O, P, Q, R, S, T, U, V, W, X, Y, Z,), (, / , *, ascii, \,} \end{array} \right\}$$

$$P = \left\{ \begin{array}{l} \langle \text{SENTENCIA} \rangle ::= \langle \text{CREATE} \rangle | \langle \text{DROP} \rangle | \langle \text{LIST} \rangle | \langle \text{DISPLAY} \rangle | \\ \quad \langle \text{ADD} \rangle | \langle \text{DELETE} \rangle | \langle \text{INSERT} \rangle \\ \langle \text{PREPARE} \rangle | \langle \text{EXTRACT} \rangle | \langle \text{REPLACEALL} \rangle | \langle \text{REMOVE} \rangle \\ \langle \text{CREATE} \rangle ::= \text{create corpus } \langle \text{ID} \rangle ; \\ \langle \text{DROP} \rangle ::= \text{drop corpus } \langle \text{ID} \rangle ; \\ \langle \text{LIST} \rangle ::= \text{list } * \text{ from } \langle \text{ID} \rangle ; \\ \langle \text{DISPLAY} \rangle ::= \text{display } \langle \text{ID} \rangle . \langle \text{NOMARCH} \rangle ; \\ \langle \text{ADD} \rangle ::= \text{add } \langle \text{DICCIONARIOS} \rangle \text{ from } \langle \text{NOMARCH} \rangle ; | \\ \quad \text{add } \langle \text{DICCIONARIOS} \rangle \{ (\langle \text{STRING} \rangle \backslash ,) \} ; \\ \langle \text{DELETE} \rangle ::= \text{delete } \langle \text{DICCIONARIOS} \rangle \text{ from } \langle \text{NOMARCH} \rangle ; | \\ \quad \text{delete } \langle \text{DICCIONARIOS} \rangle \{ (\langle \text{STRING} \rangle \backslash ,) \} ; \\ \langle \text{INSERT} \rangle ::= \text{insert } \langle \text{ID} \rangle . \langle \text{NOMARCH} \rangle ; \\ \langle \text{PREPARE} \rangle ::= \text{prepare } \langle \text{ID} \rangle . \langle \text{NOMARCH} \rangle \langle \text{USING} \rangle \\ \quad \text{put } \langle \text{NOMARCH} \rangle ; \\ \langle \text{USING} \rangle ::= (\langle \text{CICLOS} \rangle) | \langle \text{DICCIONARIOS} \rangle \\ \langle \text{CICLOS} \rangle ::= \langle \text{DICCIONARIOS} \rangle \{ \langle \text{DICCIONARIOS} \rangle \text{ and} \} \\ \langle \text{EXTRACT} \rangle ::= \text{extract } \langle \text{STRING} \rangle \text{ to } \langle \text{STRING} \rangle \langle \text{ID} \rangle . \\ \quad \langle \text{NOMARCH} \rangle \text{ put } \langle \text{NOMARCH} \rangle ; \\ \langle \text{REPLACEALL} \rangle ::= \text{replaceall } \langle \text{STRING} \rangle \text{ for } \langle \text{STRING} \rangle \langle \text{ID} \rangle . \\ \quad \langle \text{NOMARCH} \rangle \text{ put } \langle \text{NOMARCH} \rangle ; \\ \langle \text{REMOVE} \rangle ::= \text{remove acutes } \langle \text{ID} \rangle . \langle \text{NOMARCH} \rangle \text{ put } \\ \quad \langle \text{NOMARCH} \rangle ; \\ \langle \text{ID} \rangle ::= \langle \text{LETRA} \rangle \{ \langle \text{LETRA} \rangle \backslash | \langle \text{DIGITO} \rangle \} \\ \langle \text{LETRA} \rangle ::= \text{a| b|c...z|A|B|C...Z} \\ \langle \text{DIGIT} \rangle ::= \text{0| 1|2...9} \\ \langle \text{NOMARCH} \rangle ::= \langle \text{ID} \rangle . \langle \text{ID} \rangle \\ \langle \text{STRING} \rangle ::= \langle \text{LETRA} \rangle \{ \langle \text{LETRA} \rangle \} | \langle \text{DIGITO} \rangle \{ \langle \text{DIGITO} \rangle \} | \\ \quad \langle \text{DIGITO} \rangle \{ \langle \text{LETRA} \rangle \} | \langle \text{SIMBOLO} \rangle \{ \langle \text{LETRA} \rangle | \\ \quad \langle \text{SIMBOLO} \rangle \} \\ \langle \text{DICCIONARIOS} \rangle ::= \text{STOPWORDS} | \text{SYMBOLS} | \{ \text{LABELS} \} \\ \langle \text{SIMBOLO} \rangle ::= \text{ascii} \end{array} \right\}$$

Después de presentar el diseño del lenguaje, en la siguiente sección procedemos a mostrar la implementación del mismo.

3 Implementación de la gramática

3.1 Implementación en Java

Se ha elegido a Java como lenguaje de programación para la implementación de la gramática y del lenguaje en general por varias razones. Entre las más importantes se tienen las siguientes: 1) Está diseñado para ser independiente de la plataforma y 2) proporciona clases para el manejo de expresiones regulares. Con el uso de las expresiones regulares se tiene un amplio abanico de posibilidades principalmente para hacer búsquedas, para sustituir ocurrencias y para comprobar formaciones

de cadenas. Con la inclusión de las expresiones regulares, Java se convierte en un lenguaje de programación tan flexible como otros más tradicionales: *awk*, *perl*, etc. El lenguaje de preprocesamiento cuenta con un intérprete de comandos. Este programa tendrá inmerso un analizador léxico, sintáctico, semántico y por último un cargador de operaciones primitivas.

El analizador léxico lee los caracteres uno a uno desde la entrada y forma grupos de caracteres con alguna relación entre sí denominados *tokens*, que constituirán la entrada para la siguiente etapa del lenguaje; algunos *tokens* válidos para el lenguaje son: **prepare**, **drop**, **in**, **put**, **from**, **insert**, **punto**, **coma**, etc. El analizador sintáctico, toma como entrada los *tokens* que recibe del analizador léxico y comprueba si éstos llegan en orden correcto (orden con respecto a la sintaxis del lenguaje). A continuación se muestran algunos ejemplos de construcciones sintácticas válidas y aceptadas por el lenguaje de preprocesamiento:

```
create corpus trec;  
add stopwords from temp.txt;  
insert trec . ism001.txt;  
display trec . introduccion.txt;  
list * from trec;  
prepare trec . ism001.txt using(stopwords and symbols)  
put salida.txt;  
drop corpus trec;  
remove all acutes trec . uno.txt put temp.txt;
```

La función que realiza el analizador semántico es verificar que las sentencias tecleadas por el usuario tengan significado en el lenguaje y no sean un absurdo. Por último, la función del cargador de funciones es colocar en memoria las colecciones de palabras cerradas, símbolos y etiquetas cada vez que se invoca al lenguaje de preprocesamiento.

El haber programado la gramática en el lenguaje Java, proporcionó un mayor control sobre el código generado, sin embargo se invirtió una mayor cantidad de tiempo para implementar las técnicas de compilación.

3.2 Implementación en Flex y Bison

Existe un par de herramientas que hacen más fácil el desarrollo de compiladores a partir de la especificación de los lenguajes en expresiones regulares y de reglas BNF. Estos programas, llamados *flex* y *bison*, fueron utilizados para comprobar si la gramática fue construida correctamente.

Se tradujo la gramática diseñada en el formato requerido por *flex* y la salida obtenida de éste fue usada como archivo de entrada para *bison*. El programa *bison* a su vez generó un archivo de salida con código fuente en lenguaje C que, después de ser compilado, generó un programa con la capacidad de verificar la sintaxis del lenguaje de preprocesamiento.

4 Pruebas

4.1 Corpora de prueba

A continuación se muestra una descripción detallada de las características que tienen los *corpora* utilizados como recursos de prueba.

Corpus TREC-5: Es una colección de la agencia de noticias del diario de Guadalajara Jal. formada por un total de 58,062 documentos (198 MB).

Corpus EuroGOV: Está compuesto de documentos Web obtenidos de sitios gubernamentales Europeos. El *corpus* tiene cerca de 3 millones y medio de documentos en diferentes idiomas provenientes de diversos países o dominios de Internet (fr, es, uk, pt, etc). El tamaño aproximado es de 70 Gigabytes.

Corpus Medicina: Textos del área de Medicina, compilados en un solo archivo de 5.17MB, construido por un alumno de la FCC Benemérita Universidad Autónoma de Puebla.

Corpus Economía: Documentos del área de Economía, compilados en un solo archivo de 3.18MB, construido por un alumno de la FCC Benemérita Universidad Autónoma de Puebla.

Estas dos últimas colecciones carecen de formatos complejos, únicamente poseen un delimitador simple entre documentos.

4.2 Ejemplos de prueba

Las figuras 2 y 4 muestran los resultados arrojados después de preprocesar los *corpora* TREC-5 y EuroGOV (se usa el programa generado en java).

A continuación se muestra el conjunto de sentencias introducidas por el usuario para realizar el preprocesamiento del *corpus* TREC-5, EuroGOV.

```
create corpus trec;
insert trec . ism001.txt;
list * from trec;
display trec . introduccion.txt;
prepare trec . ism001.txt using(stopwords and symbols and labels)
put salida.txt;
```

```
create corpus eurogov;
insert eurogov . in.txt;
list * from eurogov;
display trec . in.txt;
remove acutes eurogov . in.txt put out.txt;
prepare eurogov . out.txt using stopwords
put final.txt;
```

```

<DOC>
<DOCNO> SP94-0202041 </DOCNO>
<ARTNUM> 0202041 </ARTNUM>
<HEADLINE>
  Para viajar en avión
</HEADLINE>
<TEXT>
hay que tener buen oído
El NORTE / Especial
  Como consecuencia de los cambios en la presión atmosférica que llevan
  aparejados, los viajes en avión pueden producir serias alteraciones en el oído
  medio, sobre todo si estaba previamente lesionado.
  En condiciones normales, gracias a la compensación que se produce
  mediante la entrada de aire a través de la trompa de Eustaquio ésta comunica el
  oído medio con la nasofaringe la presión que existe a ambos lados del tímpano es
  expandirse.
</TEXT>
</DOC>

```

Figura 1. Corpus TREC-5 sin preprocesar

```

Para viajar avión
tener buen oído
NORTE / Especial
  Como consecuencia cambios presión atmosférica llevan
  aparejados viajes avión pueden producir serias alteraciones oído
  medio sobre todo si estaba previamente lesionado
  condiciones normales, gracias compensación produce
  mediante entrada aire través trompa Eustaquio comunica
  oído medio nasofaringe presión existe ambos lados tímpano
  expandirse

```

Figura 2. Corpus TREC-5 preprocesado

```

<P><FONT FACE="Arial">1. Con un adelantamiento de antelación a la finalización
del plazo establecido para la prestación del servicio universal
en una determinada zona, el Ministerio de Fomento realizará una
consulta pública, para determinar si existen operadores interesados
en prestarlo y en qué condiciones. A estos efectos, dichos operadores
comunicarán al Ministerio de Fomento el ámbito territorial,
período y condiciones en que estarían dispuestos a llevarlo
a cabo.</FONT></P>
<P><FONT FACE="Arial">En las zonas en las que ningún operador manifieste
su interés en prestar el servicio, se aplicará de aplicación
lo establecido en el <A HREF="#a20">artículo anterior</A></FONT>
<li>ISLAS ATLÁNTICAS DE GALICIA
</li>En fase de elaboración.</li></li>
</P>

```

Figura 3. Corpus EuroGOV sin preprocesar

```

<FONT FACE="Arial">1. adelantamiento antelación finalización
plazo establecido prestación servicio universal
determinada zona, Ministerio Fomento realizará
consulta pública, determinar existen operadores interesados
prestarlo condiciones, efectos, dichos operadores
comunicarán Ministerio Fomento ámbito territorial,
período condiciones estarían dispuestos llevarlo
a cabo.</FONT>
<FONT FACE="Arial"> zonas ningún operador manifieste
interés prestar servicio, aplicación
establecido <A HREF="#a20">artículo anterior</A></FONT>
<li>ISLAS ATLÁNTICAS DE GALICIA
fase elaboración.</li></li>

```

Figura 4. Corpus EuroGOV preprocesado

5 Conclusiones

Hemos presentado una nueva alternativa para el preprocesamiento de textos. La herramienta desarrollada permitirá al investigador de PLN evitar programar procedimientos de textos *ad-hoc* para sus *corpora*. Tal vez el mayor aporte de este trabajo sea el hecho de aplicar procedimientos estándar para la etapa de preprocesamiento, facilitando de esta manera el análisis comparativo de las técnicas de PLN, sin importar el investigador que lo realice.

Referencias

1. Pinto David y Jiménez Héctor *Recuperación de Información*, Notas de Academia (2003).
2. Reyes Berenice, Moyolt Edgar, Jiménez Héctor *Reducción de términos índice usando el punto de transición*, Benemerita Universidad Autónoma de Puebla (2003).
3. Fragueta Liset y Jiménez Héctor *Uso de lattices para la Recuperación de términos*, Benemerita Universidad Autónoma de Puebla, Puebla, México (2003).

Algorithms

Algoritmos

Genetic Algorithm for Designing Artificial Neural Networks in Time Series Forecasting with Intrinsic Mode Functions

V. Landassuri-Moreno, J. Figueroa-Nazuno

Centro de Investigación en Computación
Instituto Politécnico Nacional
México DF, Col. Lindavista, C.P. 07738
victorm@sagitario.cic.ipn.mx, jfn@cic.ipn.mx

Abstract. For Time Series forecasting, we use a method called Empirical Mode Decomposition (EMD), which is adaptive and highly efficient at identifying embedded structures. EMD allows the decomposition of one-dimensional signal into intrinsic oscillatory modes. The components, called Intrinsic Mode Functions (IMF), give more information for forecasting in Artificial Neural Networks (ANNs). Genetic Algorithm (GA) is a search process based on the laws of natural selection and genetics; it has processes of *Selection*, *Mutation* and *Crossover* to reach a good solution. We apply GA to find the optimal structures (topologies) of ANNs; here the ANN (phenotype) is encoded to form the genome (genotype). The encoding process was made using Direct Encoding. The results show that ANNs with IMF-like inputs have better performance than ANNs with raw data-like inputs.

1 Introduction

Artificial Neural Networks have been used to forecast Time Series (TS), but the search of optimal structure takes a lot of time and sometimes we do not find a good structure that could give a better result. For this reason in this paper we use Genetic Algorithm (GA) to automatically find the optimal topology of ANNs. The phenotype is encoded as a bit string. Direct Encoding Scheme (DES) is used to encode the phenotype into genotype; DES specifies in the genome every connection and node that will appear in the phenotype.

The representation of an ANN is composed of four matrices (Inputs, Bias Connect, Layer Connect, and Nodes per Layer). Each row of the matrix is concatenated, to form a bit string (a part of the genome). Then GA has four inputs, each one will be processed with his basic operations (Selection, Mutation, and Crossover). The GA has to evaluate every member of the population with one fitness function, taking every genome (every member of the population) and decoding it to form the phenotype (the four matrices that represent the ANN). When we have the phenotype, we can train the network and measure its performance. In this paper we used the Average Error (AE)- and Maximum Error (ME)-like fitness. EMD is a new nonlinear technique that gives

more information on a signal. It can be a useful time series analysis tool; for this reason we apply EMD-like inputs in the ANNs to obtain a good forecasting.

The comparison in this paper is made between IMF and raw data-like inputs to the ANN designed by a GA, with AE and ME obtained from the different test sets and simulated ANN.

This paper is organized as follows. In Section 2 we describe the implementation of GA with ANNs. In Section 3 we present the EMD. In Section 4 we show the result of the computer simulation. Finally, we give the conclusions in Section 5.

2 Genetic Algorithm for Designing ANNs Topologies

GA is a method for solving optimization problems. It is based on natural selection, simulating the biological evolution. A GA at each step modifies a population. It selects individuals (parents) at random from the current population and uses them to produce children for the next generation. Since the algorithm gives preference to best parents, after several generations the population "evolves" toward an optimal solution.

To evolve ANNs we have to encode the phenotype. Here we use Direct Encoding scheme: we specify in the genome every connection and node that will appear in the phenotype (binary matrices). Figure 1 show the cycle of the GA to design ANNs.

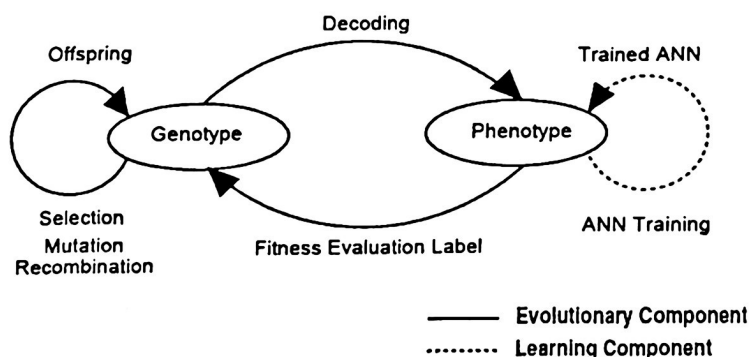


Fig. 1. Evolutionary Design of Neural Networks [3].

2.1 Parameters of the Genetic Algorithm

The GA uses three main types of rules at each step to create the next generation: *Selection* rules choose parents for the next generation basing on their fitness; *Crossover* rules combine two parents to form children for the next generation; *Mutation* rules apply random changes to individual children.

There are other functions such as the following. *Fitness scaling* converts the raw fitness scores (the value returned by the fitness function) into values in a range that is suitable for the selection function. *Reproduction* specifies how the genetic algorithm creates children for the next generation; it can be Elite fraction, which specifies the number of individuals that are guaranteed to survive to the next generation or Cross-over fraction, which specifies the fraction of the next generation produced by cross-over.

There are some stopping criteria to finish the GA, such as the following: Generations—to stop when the maximum number of iterations is reached; Time limit—to stop the algorithm after a certain time running. Fitness limit: if GA reaches a certain fitness value, etcetera. Our stopping criteria were set to 100 generations with 20 individual from each population; the population type is bit string, the Fitness Scaling is Rank function, the Selections is Stochastic uniform function, the Mutation is Gaussian function, and the Crossover is Scattered function.

2.2 ANN's Parameters

When the GA has to evaluate a member of the population, it calls the Fitness Function, takes the genome and decodes it in the phenotype (ANN). Then GA trains the Network to obtain the fitness value. When this is finished, the ANN is simulated with a set of data not previously seen by the ANN (test set). We used AE and ME to estimate the fitness, this value is returned to GA for continuation of the evolution. The number of epochs used in the training phase is varied because we do not know the current size of the network and if the epochs are too numerous, we will have over-fitting, or otherwise, the ANN will not reach its best performance because of too few epochs.

The mean square error (MSE) of the ANN has one problem: in case of over-fitting the MSE is close to zero, and the forecasting will be wrong, because the network learned the training set. For this reason we decided to use the Average Error and Maximum Error as others fitness measures. The training algorithm for the ANNs is the Levenberg-Marquardt algorithm; with Linear Transfer Functions in output layers and Tan-Sigmoid in hidden layers. The genome consists of four variables representing Inputs, Bias Connections, Layer Connections, and Nodes per Layer; with this we can build the ANN to train it.

2.3 Encode & Decode

Our phenotype consists of four matrices, which represent the network (Direct Encoding Scheme); all lines of a matrix are concatenated to form variables of the genome, see Fig. 2.

The first matrix of the phenotype is the Inputs, see Fig. 2, where its size is M_{input} (layers by number of input variables). The second is Bias Connections that represents whether or not the layer has a bias, $M_{biasconnect}$ (layers by 1). The third is Layer Connections that represent the connection between nodes $M_{layerconnect}$ (layer by layer). The fourth is the Nodes per Layer $M_{node/layers}$ (layer-1 by 3); 3 bits to permit

maximum of seven nodes per layer. The lines of these four matrices are concatenated, each one representing one variable. These are introduced as the genome of GA. In the matrix Mnode/layer we put three bits to restrict the size of the network; its size is layer-1 by 3.

Theoretical works which show that a single hidden layer is sufficient for ANNs to approximate any complex nonlinear function with any desired accuracy [6, 8]. Most authors use only one hidden layer to forecast. However, one-hidden-layer networks may require a very large number of hidden nodes, which is not desirable since the training time and the network generalization ability will degrade. Two-hidden-layer networks may provide more benefits for some type of problems [1]. Several authors address this problem and consider more than one hidden layer (usually two hidden layers) [16]. Here we permit more hidden layers.

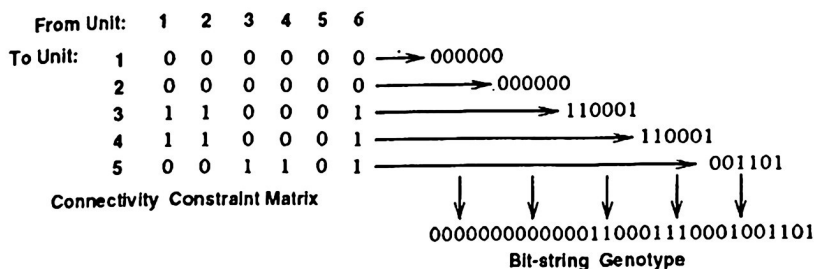


Fig. 2. Encoding of genotype from phenotype (Input matrix). The 1's represent connections between inputs variables to nodes, e.g., (3, 1) stands for connection from variable 1 to node 3, and (3, 6) for connection from variable 6 to node 3.

3 Empirical Mode Decomposition

The Empirical Mode Decomposition was introduced by Huang *et al.* in 1998 [12]. The idea of this technique is to divide a signal $x(t)$ into a sum of functions satisfying an assumption: they have at least two extrema, one maximum and one minimum. The method essentially involves two steps:

1. Two smooth splines are constructed connecting all the maxima of $X(t)$, respectively, to get its upper envelope $X_{\max}(t)$ and $X_{\min}(t)$. The extrema are found by determining the change of sign of the derivative of the signal. All the data points should be covered by the upper and lower envelopes.
2. The mean of the two envelopes is subtracted from the data to get a difference signal $X_1(t)$,

$$X_1(t) = X(t) - \frac{X_{\max}(t) + X_{\min}(t)}{2} \quad (1)$$

While it satisfies the criteria of an intrinsic mode function, this process is repeated. The first IMF $C_1(t)$ is obtained after a certain number of iterations limited for standard

deviation (SD), computed for two consecutives shifting results. In our experiments SD is set to 0.3. The original signal can be reconstructed using the following equation where r_n is the residue.

$$X(t) = \sum_{j=1}^n C_j(t) + r_n(t) \quad (2)$$

Another way to explain how the empirical mode decomposition works is as follows: it picks up the highest frequency oscillation that remains in the signal. Thus, locally, each IMF contains lower frequency oscillations than the one extracted before. This property can be very useful to pick up frequency changes; since a change will appear even more clearly at the level of an IMF [10]. Fig. 3 shows one signal with its eight IMFs.

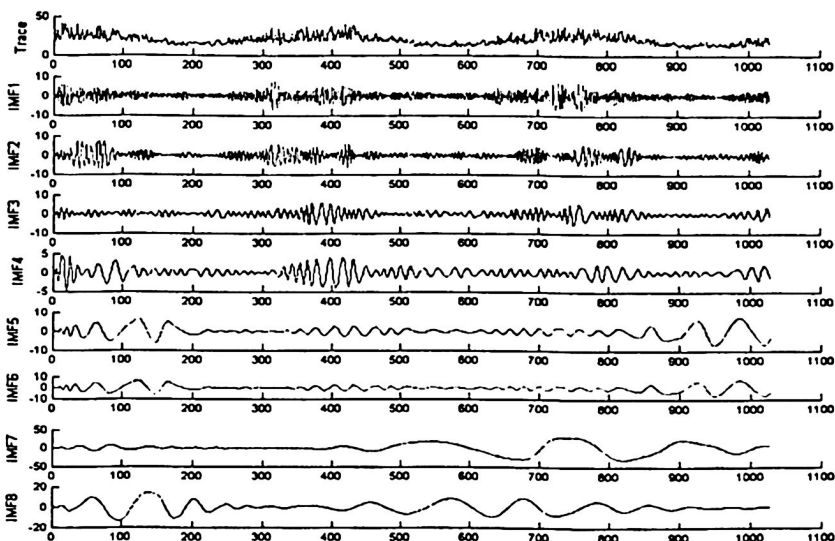


Fig. 3. Daily Maximum Temperatures in Melbourne, original data and IMF 1–8 from Australia, 1981–1990. (Source: Australian Bureau of Meteorology.)

4 Results

All ANNs of the GA were trained to forecast one step ahead. We used some Time Series to prove the performance of GA to design ANNs with IMF and without it (raw data). The Time Series were:

Meteorology: Daily Maximum Temperatures in Melbourne (DMaxTM); Daily Minimum Temperatures in Melbourne (DMinTM); Daily Precipitation, Hveravellir (DPH); *Micro-Economic*: Daily Morning Gold Prices (DMGP). *Hydrological*: Annual Minimum Level of Niel River (AMinLNR); Water Level of Corpus Christi, Texas, Estuary (WLCC).

All the time series have one variable, the only case is WLCC that have six variables (pressure, wind speed, wind direction, significant wave height, Dominant wave period and mean wave directions). This time series forecasting has the best performance, with smallest AE and ME.

In some cases with the AE, the performance of the network was very good, but the Series forecast had many jumps, crossing the real data (test set); this causes many partial errors was zero or close to zero, and when all errors were averaged, the AE was very small with respect to all variations.

In Tables 1 and 2 we show the results obtained with AE, and in Tables 3 and 4 the results with ME, e.g., of network's format: fb[7 3 1] is a feedback network with three layers and seven nodes in the first layer, three nodes in second layer and one node in the output layer; if the network does not have the suffix "fb," this means that it is a feed-forward network.

The matrices of Layer connections are represented as in Fig. 2, where (3, 1) represents the connection of the layer 1 to 3, or (1, 3) represents the connection of layer 3 to 1 (feedback); in the following matrices, every line of the matrix is separated by semicolon. The best network obtained is marked in boldface.

Table 1. Result of simulation with IMF and Average Error.

Time Serie	AE	Network	Bias Connect	Layer Connect	Epochs
DMaxTM	2.6022	fb[6 4 1]	[1;1; 1]	[000;110;110]	42
DMinTM	1.5024	fb[2 2 1]	[1;1; 1]	[001;110;010]	60
DPH	1.9534	fb[7 7 1]	[1;1;1]	[000;110;110]	59
DMGP	3.3267	fb[4 1]	[1;1]	[01;10]	68
AMinLNR	0.2998	fb[7 1]	[1;1]	[00;11]	57
WLCC	0.0582	fb[4 3 1]	[1;0;1]	[001;100;101]	81

Table 2. Result of simulation with Raw data and Average Error.

Time Serie	AE	Network	Bias Connect	Layer Connect	Epochs
DMaxTM	4.3116	fb[7 3 1]	[1;1;1]	[001;100;010]	113
DMinTM	1.8775	fb[6 4 1]	[1;0;1]	[000;110;110]	41
DPH	2.7118	fb[5 1]	[1;1]	[01;10]	100
DMGP	2.5932	fb[7 3 1]	[1;1;0]	[001;100;010]	6
AMinLNR	0.6929	fb[7 1]	[1;1]	[01;10]	109
WLCC	0.0578	[6 1]	[1;1]	[00;10]	250

Table 3. Result of simulation with IMF data and Maximum Error.

Time Serie	ME	Network	Bias Connect	Layer Connect	Epochs
DMaxTM	8.3562	fb[4 3 1]	[1;1;1]	[001;100;101]	11
DMinTM	3.5664	fb[6 4 1]	[1;1;1]	[000;110;110]	15
DPH	17.851	[7 1]	[1;1]	[00;10]	100
DMGP	11.783	fb[4 1]	[1;1]	[01;10]	47
AMinLNR	1.3782	fb[6 4 1]	[1;1;0]	[000;110;110]	21
WLCC	0.2366	fb[4 3 1]	[1;0;1]	[001;100;101]	11

Table 4. Result of simulation with Raw data and Maximum Error.

Time Serie	ME	Network	Bias Connect	Layer Connect	Epochs
DMaxTM	15.804	[4 4 1]	[1;1;1]	[000;100;010]	100
DMinTM	5.9309	[7 1]	[1;0]	[00;10]	250
DPH	21.6	[7 1]	[1;1]	[00;10]	150
DMGP	8.9345	fb[7 6 3 1]	[1;1; 1;1]	[0000;1010; 0100;0010]	200
AMinLNR	2.3641	fb[2 2 1]	[1;1;1]	[001;110;010]	17
WLCC	0.3227	fb[2 2 1]	[1;1]	[001;110;010]	50

Next, we presented the results obtained for the Annual Minimum level of Niel River. As can be observed, without IMF the network cannot give a good forecasting, see Fig. 4. Here IMF and AE slightly outperform Fig. 5. In fig. 6 the best forecasting obtained with IMF and ME is shown.

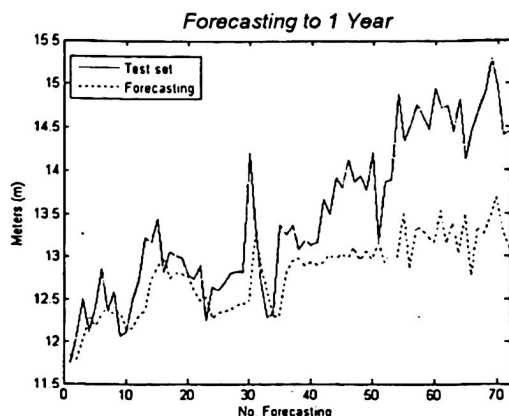


Fig. 4. Forecasting with raw data and AE = 0.6929 (1 step ahead).
Annual Minimum level of Niel River.

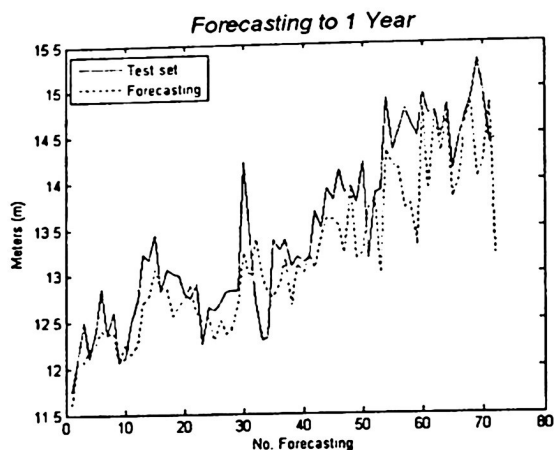


Fig. 5. Forecasting with IMF data and $AE = 0.2998$ (1 step ahead). Annual Minimum level of Niel River.

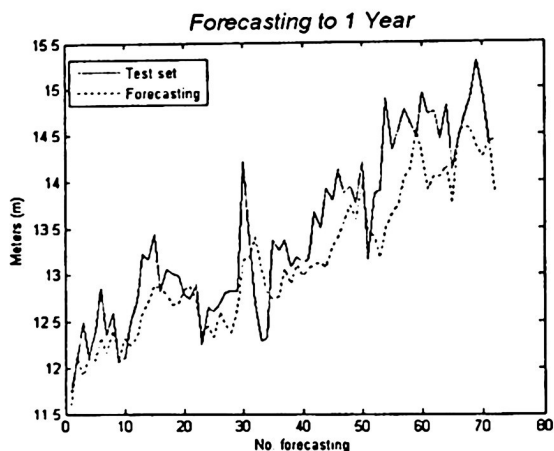


Fig. 6. Forecasting with IMF data and $ME = 1.3782$ (1 step ahead). The better forecasting for Annual Minimum level of Niel River.

Fig. 7 presents the best forecasting for WLCC whit raw data and AE, the experiment with IMF and ME are too similar, but the complexity of the network was very high. Thus we consider the best the data of the Fig 7.

We can see four TS with IMF, which give the best performance (DMaxTM, DMinTM, DPH, and AMinLNR) and the others two, with raw data (WLCC and DMGP). However, we can consider that WLCC is better with IMF, too, because the results are very similar for IMF and raw data. In general IMF gives good results.

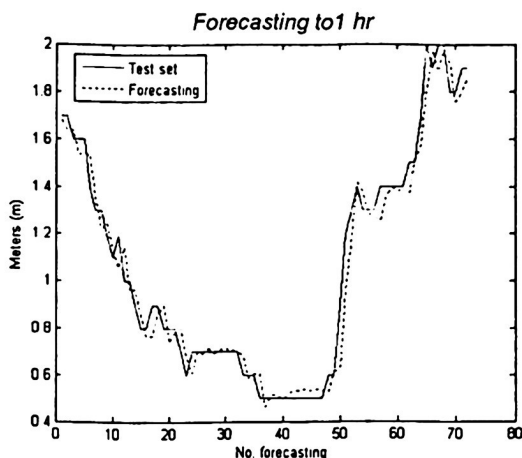


Fig. 7. The better forecasting with Raw data and AE = 0.0578 (1 step ahead).
Water Level for the Corpus Christi, Texas, Estuary.

5 Conclusions

As our results show, IMF data as input to the network is better than raw data. However, as in Water level of Corpus Christy (WLCC) example, the network with raw data and AE provides the best forecasting. If we have a sufficient number of variables that affect, directly or indirectly, the phenomena that we want to predict, then we will have a better forecasting. The only problem is the following: when the signal is divided in its Intrinsic Mode Functions, the variables are too numerous for the network. As in WLCC with IMF and ME, the forecasting is very similar, but the complexity of the network was larger than the network with raw data and AE.

In DMGP the better forecast was obtained by the network with raw data and ME, comparing with the network of IMF and AE. It has bad predictions in the last points. We conclude that dynamic time series can affect the networks and become a problem to find a good structure with GA.

The networks obtained by the GA were training with the same test data with 3, 6, 12, 24, 48, and 72 steps ahead; but the error was worse than the one obtained with one step ahead. This means that the GA can find a good network for dynamic time series presented, and the network cannot learn a different dynamics. If we want time series forecasting with another step ahead, we have to run the GA again. We can apply IMF to improve forecasting if we have a small group of variables of the phenomena, but it is also convenient to use some measures of fitness. We can see in the results that GA with ANNs is a good tool to find structures, and using IMF provides better forecasting in some cases.

References

1. Barron, A.R.: A comment on "Neural networks: A review from a statistical perspective". *Statistical Science* 9 (1) (1994), 33–35.
2. Belew, R. McInerney, J., Shraudolph, N.: *Evolving Networks*. CSE Technical Report #CS90-174. Cognitive Computer Science Research Group June (1990), Computer Science & Engr. Dept. (C-014)
3. Blakrishnan, K., Honovar V.: *Evolutionary Design of Neural Architectures – A preliminary Taxonomy and Guide to Literature*. CS TR # 95-01. Department of Computer Science. Iowa State University (1995), <http://citeseer.ist.psu.edu/balakrishnan95evolutionary.html>.
4. Branke, J.: *Evolutionary Algorithms for Neural Networks Design and Training*. Technical Report No. 322, University of Karlsruhe, Institute AIFB.
5. Coughlin, K., Tung, K. 2004a.: 11-year solar Cycle in the stratosphere extracted by the empirical mode decomposition method. *Adv. Space Res.*, 34, 323–329.
6. Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Mathematical Control Signals Systems* 2 (1989), 303–314
7. Flandrin, P., Rilling and Goncalves, P.: Empirical Mode Decomposition as a filter bank. *IEEE Sig. Proc Lett.*, (2003)
8. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. *Neural Networks* 2 (1989), 359–366
9. Kuri, A.: *A Comprehensive Approach to Genetic Algorithms in Optimization and Learning, Theory and Applications*. Vol 1. Foundations, Colección de Ciencia de la Computación, Centro de Investigación en Computación, Instituto Politécnico Nacional. México (1999)
10. Magrin, I., Baraniuk, R.: *Empirical mode decomposition based time–frequency attributes*, Rice University, Houston Texas (1999)
11. McCluskey, P.: *Feedforward and Recurrent Neural Networks and Genetics Programs for Stock Market and Time Series Forecasting*. Department of Computer Science, Brown University. CS-93-36
12. N. Huang., Shen, Z., Long, S., Wu M., Shih H., Zheng Q., Yen N., Tung C., and Liu H.: The empirical mode decomposition and hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. of the Royal Society of London* vol. 454 (1998), pp. 903–995
13. Stanley, K., & Miikkulainen, R.: Efficient Evolution of Neural Networks Topologies. *Proceedings of the 2002 Congress on Evolutionary Computation (CEC '02)*. Honolulu, Hawaii: IEEE
14. Yao, X., & Liu, Y. A New Evolutionary System for Evolving Artificial Neural Networks, *IEEE Transactions on Neural Networks*, 8(3):694–713, May (1997)
15. Yao, X. A Review of Evolutionary Artificial Neural Networks. *International Journal of Intelligent Systems*, 8:539–567
16. Zhang, G., Patuwo, B., Hu, M.: Forecasting with artificial neural networks: The state of the art. *International journal of Forecasting* (1998). 14 35–62 N.H Elsevier
17. The MathWorks web site. <http://www.MathWorks.com>.
18. Time Series Data Library. <http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/>

Método y algoritmo de árbol de cubos y sus aplicaciones para resolver diferentes problemas de optimización discreta

Vladimir Khachaturov¹, José Crispín Zavala Díaz²

¹ Profesor-Investigador, Facultad de Ciencias-UAEM,
Cuernavaca, Morelos, México
vr_khach@yahoo.com

² Profesor-Investigador, Facultad de Contabilidad, Administración e
Informática-UAEM, Cuernavaca, Morelos, México
jc_zavala2002@yahoo.com

Resumen. Presentamos un método para solucionar problemas de optimización discreta, que llamamos "el árbol de cubos". El método se fundamenta en la teoría de conjuntos y en la teoría de latices. El método se puede utilizar para analizar problemas de optimización discreta de grandes dimensiones porque está estructurado de tal forma que fácilmente se puede paralelizar. Para ejemplificar su aplicación se muestra la solución del problema de la selección óptima del subconjunto de objetos. Los resultados de los experimentos en un procesador de cientos de tareas nos enseñan que el número de cálculos no fue más que m^4 . Nuestras valoraciones nos muestran que para 32 procesadores la dimensión de la tarea se podría incrementar hasta 4 veces más.

1 Planteamiento del problema

Sea el conjunto finito $I = \{1, 2, \dots, m\}$ y sea el conjunto $B(I)$ el conjunto de todos los subconjuntos, esto es: $B(I) = \{\omega \mid \omega \subseteq I\}$ y $|B(I)| = 2^m$

Al conjunto $B(I)$ es posible imaginarlo como un cubo de dimensión m , donde los vértices del cubo son subconjuntos $\omega \subseteq I$, estos subconjuntos tienen un orden parcial entre ellos dado por la teoría de conjuntos con las operaciones: $\subseteq$, $\cup$ y $\cap$.

El cubo de dimensión m lo denotamos como $C(m)$, es claro que los elementos (vértices) del cubo son subconjuntos $\omega \subseteq I$, es posible escribirlo como el intervalo:

$$C(m) = [\emptyset, I] = \{\omega \subseteq I \mid \emptyset \subset \omega \subset I\}$$

Igualmente para el intervalo $[\omega_1, \omega_2]$, donde $\omega_1 \subset \omega_2 \subset I$, corresponde a un cubo $C(k)$

$$C(k) = [\omega_1, \omega_2] = \{\omega \subseteq I \mid \omega_1 \subset \omega \subset \omega_2\}, \text{ con la dimensión } k = |\omega_2 \setminus \omega_1|.$$

Todos los elementos (vértices) de $C(k)$ son vértices de $C(m)$. Si tenemos dos cubos $C(k_1) = [\omega_1^1, \omega_2^1]$, $C(k_2) = [\omega_1^2, \omega_2^2]$ y $\omega_1^2 \subset \omega_1^1$, $\omega_2^2 \subset \omega_2^1$ entonces se puede escribir $C(k_1) \subset C(k_2)$. Para $k \leq m$: $C(k) \subset C(m)$.

En este trabajo estudiaremos la tarea de partir el cubo $C(m)$ en los conjuntos Π de cubos de dimensión menor, donde para cada dos cubos la intersección es el conjunto vacío y la unión de todos los cubos es igual a $C(m)$.

Supongamos $\Pi = \{C_1, C_2, \dots, C_r\}$, donde C_p es un cubo donde $p = 1, 2, \dots, r$. Vamos a decir que Π es la partición si:

- $C_i \cap C_j = \emptyset$ para cada $i \neq j, 1 \leq i \leq r, 1 \leq j \leq r$
- $\bigcup_{p=1}^r C_p = C(m)$

2 Procedimiento para la construcción del árbol de cubos

Sea el cubo inicial $C(m) = [\emptyset, I]$, $I = \{1, 2, \dots, m\}$, vamos a construir un medio especial con el conjunto parcialmente ordenado de cubos $C(l)$, $l=m-k$ donde $k=0, 1, \dots, m$, de diferentes dimensiones. Al nivel superior $l=m$ ($k=0$) le corresponde el cubo único $C(m)$ de dimensión m . En el nivel siguiente $l=m-1$ ($k=1$) se distribuyen dos cubos disjuntos: $C_1(m-1)$ y $C_2(m-1)$ de dimensión $(m-1)$ cada uno, estos cubos se pueden hacer de m distintas formas, nosotros hacemos lo siguiente: para el cubo $C_1(m-1)$ consideramos un intervalo $\{1\}$, I y al cubo $C_2(m-1)$ le corresponde un intervalo $[\emptyset, I \setminus \{1\}]$. Estos dos cubos son subconjuntos disjuntos, $C_1(m-1) \cap C_2(m-1) = \emptyset$, porque todos los elementos del cubo $C_1(m-1)$ tienen al subconjunto $\{1\}$ y todos los elementos del cubo $C_2(m-1)$ no tienen al mismo elemento $\{1\}$.

El número de elementos de cada uno de los cubos C_1 y C_2 es igual a $2^{(m-1)}$, entonces el número de elementos de ambos cubos es 2^m , por eso: $C_1(m-1) \cup C_2(m-1) = C(m)$. Por tanto, el conjunto de los dos cubos $C_1(m-1)$ y $C_2(m-1)$ es la partición del cubo $C(m)$. Notamos que cada cubo del nivel $l=m-k$ tiene la dimensión $l=m-k$, entonces tiene 2^l vértices.

Posteriormente se forman los niveles siguientes $(m-2)$, en cada cubo $C_1(m-1)$ y $C_2(m-1)$ se hacen las mismas operaciones. Cada cubo se parte en dos cubos de una dimensión menor, entonces en el nivel $(m-2)$ recibimos a cuatro que son una partición del cubo $C(m)$.

Si seguimos este proceso llegamos al nivel $m-3$ con ocho cubos, dos cubos del nivel $m-3$ para cada uno de los cubos del nivel $m-2$. En general para el nivel $(m-k)$, $0 \leq k \leq m$, se pueden formar 2^k cubos, donde cada dos cubos son conjuntos disjuntos y la unión de todos ellos forman el cubo inicial $C(m)$. Entonces todos los cubos del nivel $(m-k)$ es la partición del cubo $C(m)$.

En el último nivel ($l=0$) el número de cubos de dimensión cero es igual a 2^m . Entonces es igual al número de vértices del cubo inicial $C(m)$. El conjunto de ellos es la partición, porque es claro que cada intersección es igual a cero y la unión de ellos es el conjunto inicial $C(m)$.

Construimos un árbol donde sus vértices son los cubos definidos con el procedimiento descrito. En cada etapa donde recibimos cubos, en el nivel siguiente se construye un lado entre el cubo del cual nacieron con los dos cubos de una dimensión menor en uno del cubo del nivel superior. Como resultado recibimos un árbol con

vértices, donde cada vértice es un cubo y ellos son de un orden parcial definido por el lado que los conectan. Supongamos este árbol como $A(m)$, vamos a estudiar algunas propiedades del árbol $A(m)$.

3 Algunas propiedades del árbol de cubos

3.1 Estructura jerárquica

De la formalización del algoritmo de la construcción del árbol de cubos concluimos que el árbol tiene una estructura jerárquica. El árbol tiene $m+1$ niveles $l = m - k$ ($k = 0, 1, \dots, m-1, m$), en el nivel superior ($l=m$) está un sólo vértice, en los niveles siguientes el número de vértices (número de cubos) es igual a 2^k ($k = 0, 1, \dots, m-1, m$).

3.2 Número de vértices del árbol $A(m)$

El número de todos los vértices del árbol $A(m)$ es igual a: $1 + 2 + 4 + \dots + 2^m$. Entonces $|A(m)| = 2^{m+1} - 1$.

3.3 Generación de diferentes variantes de particiones del árbol $A(m)$

El conjunto de vértices de cada nivel del árbol $A(m)$ es la partición del cubo inicial $C(m)$. Entonces en cada árbol se tiene al mismo tiempo m diferentes variantes de las particiones del cubo inicial $C(m)$. Además se puede formar muchas otras variantes de la partición. Por ejemplo se puede tomar como base una variante de la dimensión que le corresponde a cualquier nivel ($l > 0$) del árbol y sustituir, dentro de esta variante base, uno de los elementos (o varios elementos) con dos cubos de dimensión menor a uno, que nosotros recibimos con el procedimiento de construcción del árbol. Esta propiedad del árbol de cubos permite organizar eficazmente los cálculos en las máquinas paralelas.

3.3 Dependencia del número de vértices del árbol conforme a su nivel

La dependencia del número de vértices del árbol $A(m)$ conforme al nivel l ($N(l)$, $l=m-k$, donde $k = 0, 1, 2, \dots, m$) está dada por:

$$N(k) = 2|x(l(k))| = 2^{m-l(k)} = 2^k$$

Donde

$l(k)=m-k$, ($k = m, m-1, \dots, 1, 0$) es el número del nivel
 $|x(l(k))| = 2^{m-l(k)} = 2^k$

el contorno de un pino que tiene un tronco que corresponde a una franja $-\frac{1}{2} \leq x \leq \frac{1}{2}$.

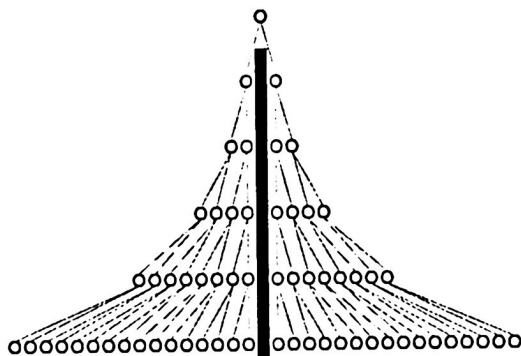


Figura 1. El árbol de cubos para $m = 5$. En el centro el tronco va de $-\frac{1}{2} \leq x \leq \frac{1}{2}$.

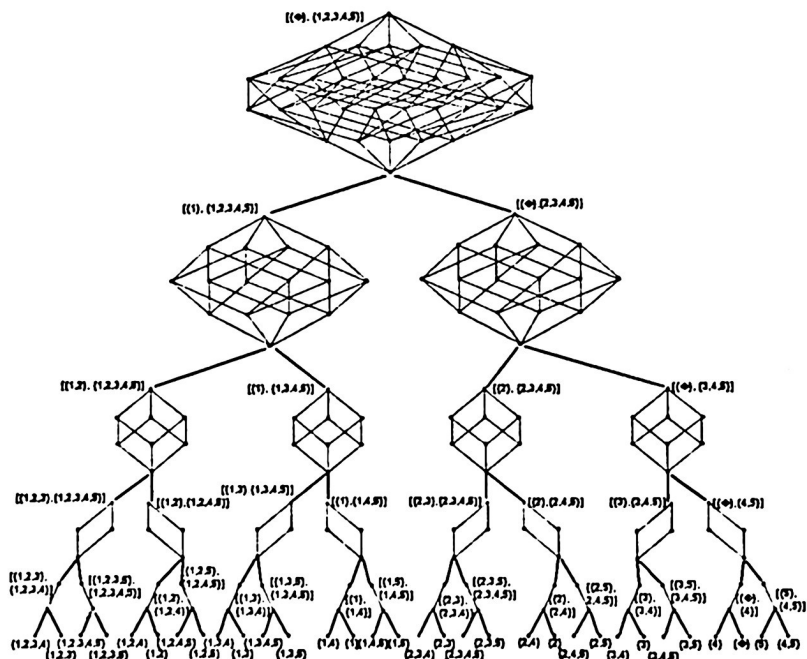


Figura 2. El árbol de cubos.

El contorno de este pino se compone de dos aproximaciones $x(l(k))=2^{k-1}$ para $x(l(k)) \geq 0$ y $x(l(k))=-2^{k-1}$ para $x(l(k)) \leq 0$. Por eso $N(x(l(k)))=2^{k-1}-(-2^{k-1})=2^k$. Notamos que si aumentamos la magnitud de m , el "pino" se desliza a lo largo del eje "l". El incremento de la capacidad de vértices del cubo sucede en los últimos niveles del "pino" nuevo.

En la figura 2 se muestra como disminuye en una dimensión los cubos de niveles consecutivos. En la parte superior el grado del cubo es de cinco, el cual se encuentra conectado a dos cubos de grado cuatro, cada uno de éstos a su vez están conectados con dos cubos de grado tres, así sucesivamente hasta el último nivel, donde los cubos son de grado cero.

3.4 Número de vértices como una función de los intervalos de los niveles del árbol

No es difícil determinar, para dos pinos con $m=m_1$ y $m=m_2$ el número de vértices del árbol de cubos en los intervalos $[m_1-k, m_1]$ y $[m_2-k, m_2]$ es el mismo, para cualquier $0 \leq k \leq \min(m_1, m_2)$ y es igual a $2^{k+1}-1$, entonces no depende del valor de m

3.5 Subárboles del árbol $A(m)$ y sus propiedades

Cada vértice del árbol forma un subárbol del árbol $A(m)$. El vértice del árbol $A(m)$ - cubo $C(m)$ - se denomina también la raíz del árbol $A(m)$, porque de él nacen todos los vértices del árbol.

Un subárbol del árbol $A(m)$ es un árbol que nació a través de cualquier vértice del árbol $A(m)$. El número de todos los subárboles es igual al número de todos los vértices del árbol, esto es igual a $2^{m+1}-1$, cada subárbol que nace de la raíz $C(m-k)$ tiene $2^{m-k+1}-1$ vértices ($k=0, 1, \dots, m$).

Entonces para $k=0$ tenemos la raíz del árbol principal $A(m)$ que tiene $2^{m+1}-1$ vértices (ver punto 3.2). Para $k=m$ cada uno de los 2^m subárboles tienen un solo vértice.

3.5.1 Sea $C^1(l_1)$ y $C^2(l_2)$, $l_2 \leq l_1$, dos raíces que corresponden a dos subárboles $A^1(l_1)$ y $A^2(l_2)$.

- Si $C^1(l_1) = C^2(l_2)$, entonces $A^1(l_1) = A^2(l_2)$.
- Si $C^1(l_1) \cap C^2(l_2) = \emptyset$, entonces $A^1(l_1) \cap A^2(l_2) = \emptyset$, que significa que los subárboles $A^1(l_1)$ y $A^2(l_2)$ no tienen vértices y lados comunes.
- Si $C^2(l_2) \subset C^1(l_1)$, entonces todos los vértices y lados del subárbol $A^2(l_2)$ también son elementos del subárbol $A^1(l_1)$.

3.5.2 Si excluimos del árbol $A(m)$ el subárbol $A(l_0)$, $l_0 < m$, excluirémos a todos los subárboles $A(l)$ con $l < l_0$, para los cuales $C(l) \subset C(l_0)$.

3.5.3 Si alguna propiedad no cumple para ningún elemento $\omega \in C(l_0)$, entonces esta propiedad no cumple para ningún elemento $\omega \in C(l)$ para todo $C(l) \subset C(l_0)$, porque $C(l_0)$ tiene todos los elementos para cualquier $C(l) \subset C(l_0)$.

3.5.4 Si alguna propiedad cumple para algún elemento $\omega_b \in C(l_0)$, entonces existe una cadena de cubos: $C^0(0) \subset C^0(1) \subset C^0(2) \dots C^0(l_0-1) \subset C^0(l_0)$, en la cual para cada $C^0(l)$, $0 \leq l \leq l_0$, existe un elemento donde esta propiedad también se cumple. Por ejemplo, la cadena de cubos para los cuales $C^0(0) = \omega_b$

3.6 El árbol de cubos como latices

Introducimos en el árbol de cubos un vértice, el conjunto vacío, y lo conectamos por medio de lados con todos los cubos del nivel cero ($l(k)=0$, $k=m$). Por el teorema siguiente determinamos que la estructura que recibimos es un lattice [1,2].

Teorema Después de introducir en el árbol de cubos el vértice que corresponde al conjunto vacío, la estructura que recibimos es una lattice.

Demostración. En correspondencia con la formación del árbol de cubos m , cada dos cubos $C_1 = [\omega_1^1, \omega_2^1]$, $C_2 = [\omega_1^2, \omega_2^2]$ y $C_1 \neq C_2$ pueden tener una de las dos propiedades:

1. $C_1 \subset C_2$, $\omega_1^2 \subset \omega_1^1$ y $\omega_2^2 \supset \omega_2^1$
2. $C_1 \cap C_2 = \emptyset$

Por el primer punto se tiene: $\supremo(C_1, C_2) = C_2$, $\infimo(C_1, C_2) = C_1$.

Por el segundo punto se tiene: el $\supremo(C_1, C_2)$ es un nuevo cubo $\tilde{C} = [\omega_1^1 \cap \omega_1^2, \omega_2^1 \cup \omega_2^2]$ si el cubo $\tilde{C} \in A(m)$, si $\tilde{C} \notin A(m)$ $\supremo(C_1, C_2) = C(m) = [\emptyset, 1]$ e $\infimo(C_1, C_2) = \emptyset$. Entonces nosotros determinamos que después de introducir el conjunto vacío en el árbol de cubos m , nosotros recibimos la estructura en la cual cualquier par de elementos siempre tienen supremo e infimo, por eso la estructura que recibimos es una lattice [1,2].

4 Método del árbol de cubos para los problemas de optimización

Con el objetivo de explicar cómo el método de árbol de cubos se aplica para determinar la solución de problemas de optimización discreta, utilizamos el planteamiento del problema de la selección optimal del subconjunto de objetos con un parámetro, que es una generalización de la tarea clásica conocida como "la tarea del maletín".

Sujeto a:
$$\sum_{j=1}^m y_j \leq b, \quad y_k \in \{0, b_{i_k}\}$$

De esta última transformación se obtienen las soluciones integral $C(I)$ y lineal $C(L)$. Dado que b_{i_k} es un entero, se tendrá un límite q tal que:

$$\sum_{k=1}^q b_{i_k} \leq b < \sum_{k=1}^{q+1} b_{i_k}$$

Porque $\sum_{k=1}^m b_{i_k} = B > \lambda_0 B = b$ entonces $q < m$ siempre. Con lo cual se obtienen los elementos que son la solución.

$$y_k = b_{i_k}, \text{ cuando } x_{i_k} = 1 \text{ para } k = 1, 2, \dots, q$$

$$y_k = 0, \text{ cuando } x_{i_k} = 0 \text{ para } k = q+1, \dots, m$$

La magnitud de la función objetivo para la solución integral estará dada por los términos donde $x_{i_k} = 1$ ó $x_{i_k} = 0$

$$C(I) = \sum_{k=1}^q c_{i_k}$$

Para la solución lineal (no integral) tenemos:

$$C(L) = C(I) + \frac{c_{i(q+1)}}{b_{i(q+1)}} \left(b - \sum_{k=1}^q b_{i_k} \right)$$

Claro que $C(L) \geq C(I)$. Puede ser $C(L) = C(I)$ sólo cuando $b = \sum_{k=1}^q b_{i_k}$. A cada solución integral, $x_{i_k} = 1$, le corresponde el subconjunto $\omega^p \subset I$, donde $i_k \in \omega^p$ para $x_{i_k} = 1$ y $i_k \notin \omega^p$ para $x_{i_k} = 0$. A cada ω^p le corresponde su $C(\omega^p)$, donde $C(I, \omega^0) = \sum_{i \in \omega^0} c_i$ y $C(L, \omega^p) \geq C(I, \omega^p)$.

Este algoritmo se puede utilizar para cualquier subárbol $A(I)$ con su raíz $C(I)$ de la manera siguiente.

Cada una de las raíces se puede representar de la forma:

$$C(I) = (\omega_1, \omega_2) = \{\omega_1 \mid \omega_1 \subset \omega \subset \omega_2\}, \text{ donde } [\omega_2 \setminus \omega_1] = I, \omega_1 \subset I \text{ y } \omega_2 \subset I.$$

Planteamiento del problema, para el cubo $C(I)$ es necesario buscar:

$$\sum_{i \in \omega_1} c_i + \max_{i \in \omega_2 \setminus \omega_1} \sum c_i x_i$$

Sujeto a:
$$\sum_{i \in \omega_2 \setminus \omega_1} b_i x_i \leq b - \sum_{i \in \omega_1} b_i x_i, \quad x_i \in \{0, 1\}, \quad i \in \omega_2 \setminus \omega_1$$

Sujeto a:
$$\sum_{j=1}^m y_j \leq b, \quad y_k \in \{0, b_{i_k}\}$$

De esta última transformación se obtienen las soluciones integral $C(I)$ y lineal $C(L)$. Dado que b_{i_k} es un entero, se tendrá un límite q tal que:

$$\sum_{k=1}^q b_{i_k} \leq b < \sum_{k=1}^{q+1} b_{i_k}$$

Porque $\sum_{k=1}^m b_{i_k} = B > \lambda_0 B = b$ entonces $q < m$ siempre. Con lo cual se obtienen los elementos que son la solución.

$$y_k = b_{i_k}, \text{ cuando } x_{i_k} = 1 \text{ para } k = 1, 2, \dots, q$$

$$y_k = 0, \text{ cuando } x_{i_k} = 0 \text{ para } k = q+1, \dots, m$$

La magnitud de la función objetivo para la solución integral estará dada por los términos donde $x_{i_k} = 1$ ó $x_{i_k} = 0$

$$C(I) = \sum_{k=1}^q c_{i_k}$$

Para la solución lineal (no integral) tenemos:

$$C(L) = C(I) + \frac{c_{i(q+1)}}{b_{i(q+1)}} \left(b - \sum_{k=1}^q b_{i_k} \right)$$

Claro que $C(L) \geq C(I)$. Puede ser $C(L) = C(I)$ sólo cuando $b = \sum_{k=1}^q b_{i_k}$. A cada solución integral, $x_{i_k} = 1$, le corresponde el subconjunto $\omega^p \subset I$, donde $i_k \in \omega^p$ para $x_{i_k} = 1$ y $i_k \notin \omega^p$ para $x_{i_k} = 0$. A cada ω^p le corresponde su $C(\omega^p)$, donde $C(I, \omega^0) = \sum_{i \in \omega^0} c_i$ y $C(L, \omega^p) \geq C(I, \omega^p)$.

Este algoritmo se puede utilizar para cualquier subárbol $A(I)$ con su raíz $C(I)$ de la manera siguiente.

Cada una de las raíces se puede representar de la forma:

$$C(I) = (\omega_1, \omega_2) = \{\omega_1 \mid \omega_1 \subset \omega \subset \omega_2\}, \text{ donde } [\omega_2 \setminus \omega_1] = I, \omega_1 \subset I \text{ y } \omega_2 \subset I.$$

Planteamiento del problema, para el cubo $C(I)$ es necesario buscar:

$$\sum_{i \in \omega_1} c_i + \max_{i \in \omega_2 \setminus \omega_1} \sum c_i x_i$$

Sujeto a:
$$\sum_{i \in \omega_2 \setminus \omega_1} b_i x_i \leq b - \sum_{i \in \omega_1} b_i x_i, \quad x_i \in \{0, 1\}, \quad i \in \omega_2 \setminus \omega_1$$

donde: $x_i = 1$ para $i \in \omega_1$, y $x_i = 0$ para $i \in \omega_2$

Buscamos $\max_{i \in \omega_1 \setminus \omega_2} \sum c_i x_i$ igual que para el cubo $C(m)$. Como resultado

recibimos, para el cubo $C(l) = C(\omega_1, \omega_2)$, las soluciones óptimas lineal $C(L, \omega^0, l)$ e integral aproximado con magnitud $C(l, \omega^0, l)$. Claro, siempre $C(L, \omega^0, l) \geq C(l, \omega^0, l)$ para $\omega_1 \subset \omega^0 \subset \omega_2$.

4.2.2 Reglas para el rechazo de subárboles no óptimos

En correspondencia con la propiedad 3.6.3, si en el cubo $C(l_0)$ no existen soluciones óptimas, entonces no hay soluciones óptimas en ningún cubo $C(l) \subset C(l_0)$. Por tanto si de alguna manera afirmamos que el cubo $C(l_0)$ no tiene soluciones óptimas, entonces no hace falta revisar todo el subárbol $A(l_0)$, en consecuencia no hace falta revisar los $2^{l_0+1} - 1$ cubos, vértices de este subárbol.

Para el problema que investigamos se pueden formular las siguientes reglas de rechazo de subárboles no óptimos.

Supongamos conocidas algunas soluciones aproximadas de nuestra tarea, las cuales están en el subconjunto $\tilde{\omega} \subset I$ con sus correspondientes magnitudes de las funciones de la solución integral $C(I, \tilde{\omega})$ y de la solución no integral (lineal) $C(L, \tilde{\omega})$, donde $C(I, \tilde{\omega}) \leq C(L, \tilde{\omega})$

Sea el subárbol $A(l)$ con su raíz $C(l) = C(\omega_1, \omega_2)$, para $C(l)$ conocemos las magnitudes de sus soluciones $C(L, \omega^0, l)$ y $C(l, \omega^0, l)$, donde $C(L, \omega^0, l) \geq C(l, \omega^0, l)$.

Regla de rechazo Si $C(I, \tilde{\omega}) \geq C(L, \omega^0, l)$, entonces se puede excluir la revisión del subárbol $A(l)$, porque entre todos los cubos que son vértices del subárbol $A(l)$ no existe ninguna solución "mejor" que $\tilde{\omega}$, es decir, no existe una solución con una significación del funcional entero mayor que $C(I, \tilde{\omega})$. La demostración de esta regla sigue la desigualdad siguiente:

$$C(I, \tilde{\omega}) \geq C(L, \omega^0, l) \geq C(l, \omega^0, l)$$

y de la propiedad de los subárboles 3.6.3.

Esta regla se utiliza permanentemente en el proceso de búsqueda de la solución óptima de la tarea inicial.

4.2.3 Algoritmo de búsqueda de la solución óptima

El algoritmo de búsqueda de las soluciones óptimas se constituye de cuatro bloques principales (etapas).

Primer Bloque. Diferentes búsquedas de las soluciones integrales aproximadas. La realización de este bloque es una etapa muy importante en la búsqueda de la solución óptima, porque la presencia de una buena solución integral permite aumentar la eficacia del algoritmo de búsqueda de la solución óptima, ya que aumenta la eficacia de la regla de rechazo.

Estos algoritmos de búsqueda deben ser bastante rápidos, por ejemplo, óptimo local, algoritmos avaros (greedy), algoritmos de búsqueda causal y otros más. Uno de estos algoritmos está descrito en el punto 4.2.1 y en el punto 4.2.4. se muestra otro método de búsqueda para las soluciones aproximadas.

Como resultados de la realización de este bloque determinamos y guardamos en la memoria la solución integral $\tilde{\omega}$ y su correspondiente significación $C(\tilde{\omega})$, la que determinamos como la solución óptima temporal.

Segundo Bloque. Esquema de selección de los vértices del árbol $A(m)$. Existen diferentes algoritmos para seleccionar los vértices de un árbol, por ejemplo las que se presentan en las referencias [4, 8, 11]. Nosotros utilizamos los algoritmos, que se han aplicado durante muchos años, para solucionar diferentes tareas de optimización combinatoria [5,6,7,9,10]. El esquema común está escrito en el punto 2 del presente trabajo.

Tercer Bloque. Realización de la regla de rechazo. Para cada vértice $C(l)=C(\omega_l, \omega_l)$ determinamos las significaciones $C(L, \omega^l, l)$, $C(l, \omega^l, l)$ y la solución integral aproximada ω^l , $\omega_1 \subset \omega^l \subset \omega_2$. Después comparamos $C(L, \omega^l, l)$ con la significación del óptimo temporal $C(l, \tilde{\omega})$.

Si $C(l, \tilde{\omega}) \geq C(L, \omega^l, l)$ excluimos del cálculo al subárbol $A(l)$, entonces no continuamos formando dos nuevos cubos del vértice $C(l)$, y pasamos a formar nuevos cubos de otro vértice en correspondencia al esquema común de formalización del árbol $A(m)$ (ver punto dos).

Si no existen otros cubos terminamos la búsqueda de la solución óptima y pasamos al bloque 4.

Cuarto bloque. Retención de las soluciones óptimas temporales y la salida del cálculo. Cada vez, después de la determinación de la solución integral, para cada vértice $C(l)$ del árbol $A(m)$ la significación de $C(l, \omega^l, l)$ se compara con la significación $C(l, \tilde{\omega})$. Si $C(l, \omega^l, l) > C(l, \tilde{\omega})$, entonces ω_l con su $C(l, \omega^l, l)$ la retenemos como la solución siguiente óptima temporal en lugar de $\tilde{\omega}$.

Después de concluir el cálculo, la última solución óptima temporal $\tilde{\omega}_u$, $C(l, \tilde{\omega}_u)$ es la solución de la tarea inicial $\omega_{opt} = \tilde{\omega}_u$, $C(l, \omega_{opt}) = C(l, \tilde{\omega}_u)$ son las salidas del algoritmo.

4.2.4 Los experimentos computacionales, versión secuencial

Los experimentos computacionales consistieron en obtener la solución integral, lineal y los subconjuntos de la solución óptima para cada λ_0 seleccionada. Se consideraron dos tipos de problemas, el primero de ellos fue determinar los coeficientes b_i y c_i por medio de una progresión aritmética, de tal forma que nosotros conocemos la suma de ellos para cualquier número de elementos, es decir la solución óptima. Una vez que se determinaron los coeficientes, éstos se distribuyeron en diferentes posiciones para dirigir la búsqueda a cubos diferentes. En todas las pruebas, el AAC determinó la solución óptima correcta, con lo cual comprobamos que el algoritmo del árbol de cubos se implementó correctamente.

El segundo experimento consistió en determinar los coeficientes (b_i , c_i) en forma aleatoria, los resultados que se presentan son para este segundo experimento. El primer experimento que se presenta para $m=500$, consiste en determinar la solución entera óptima para 20 intervalos de λ . Para este problema y el siguiente se utilizó la regla de rechazo general $C(L, \alpha^p, l) - C(I, \tilde{\omega}) \leq \varepsilon C(I, \tilde{\omega})$. Cuando $\varepsilon=0$ la regla de rechazo se utiliza para determinar la solución exacta, y cuando $\varepsilon>0$ busca la solución aproximada con un valor no mayor a $\varepsilon C(I, \tilde{\omega})$. En la figura 3 se muestra la gráfica de la solución exacta, donde en las abscisas está λ y en el eje de las ordenadas está el número de iteraciones con el cual convergió el algoritmo.

Como es de suponerse, los cálculos experimentales confirman nuestros supuestos de que los problemas más difíciles requieren de muchos cálculos, esto es cuando $\lambda=1/2$.

El número de soluciones (N) que calculamos nunca fue mayor a m' , $N < m'$. En forma indirecta nos muestra que el algoritmo que utilizamos es efectivo. Tomando como base estos valores experimentales ($N < m'$, para $\lambda=1/2$), nosotros recibimos en forma analítica los valores para otras significaciones de λ siguientes:

$$N \approx m^{x_k}, \text{ donde } x_k = 4(1 - \log_m 2^k), \quad k = 0, 1, 2, \dots$$

$$\text{para } 0 < \lambda_k \leq \frac{1}{2}, \text{ donde } \lambda_k = 2^{-(k+1)} \text{ y para } \frac{1}{2} \leq \lambda_k < 1, \text{ donde } \lambda_k = 1 - 2^{-(k+1)}.$$

Es claro que para $\lambda_k = 0$ la solución óptima es $x_i = 0$ para todos $i = 1, 2, \dots, m$, y para $\lambda_k = 1$ la solución óptima es $x_i = 1$ para todos $i = 1, 2, \dots, m$.

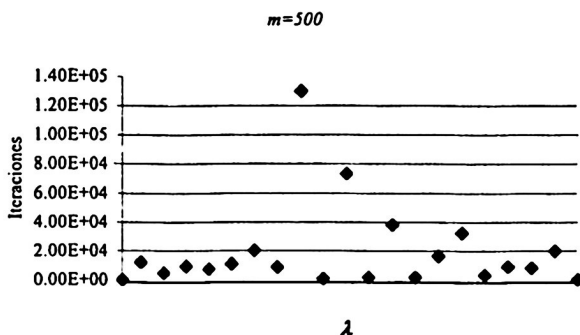


Figura 3. Iteraciones en función de λ para $m = 500$.

La magnitud de la solución óptima entera se muestra en la figura 4, como se observa la magnitud de la solución entera se incrementa monótonamente hasta alcanzar el máximo, cuando $\lambda=1$.

El segundo experimento que se presenta es para $m=3,200$ con diferentes valores de ε . Se determinan las soluciones aproximadas para $\varepsilon>0$ (6%, 4%, 2% y 1%) y la solución exacta $\varepsilon=0$. El cálculo se realizó para $\lambda = 0.0035$ a causa de que es posible obtener la solución exacta en un procesador, ya que como se mostró en el primer ejemplo, para una λ mayor y cercana a $1/2$ se requiere de un número de iteraciones

considerablemente alto. Los resultados de este segundo ejemplo se muestran en la tabla 1.

En la segunda columna de la tabla 1 están los números de iteraciones en que se calculo el valor mayor y en la tercera columna está el número de iteraciones en que se determino que este valor máximo era la solución optima. En la cuarta columna está el valor de la solución entera y en la quinta columna está el intervalo en el cual la solución optima se encuentra.

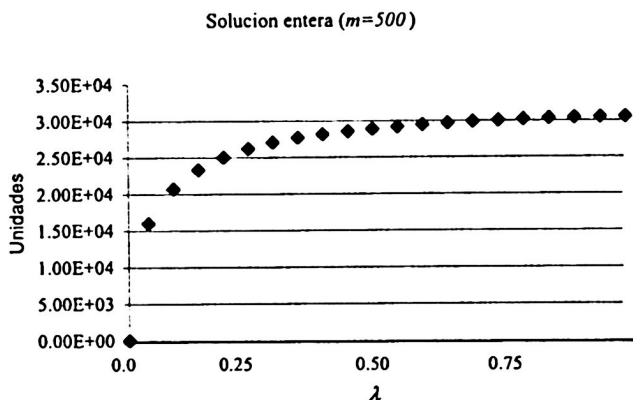


Figura 4. Magnitud de la solución entera exacta para $m=500$

El intervalo de la solución estará dado por el porcentaje considerado para obtener la solución aproximada ($\varepsilon C(I, \tilde{\omega})$). Por ejemplo para $\varepsilon=6\%$ el límite superior del intervalo es $2,202 + 0.06(2,202) = 2,334$, el cual está en la quinta columna de la tabla 1. Para el porcentaje siguiente el límite superior del intervalo es $2,401$, pero dado que la solución no será mayor al límite superior anterior, entonces consideramos éste como el límite superior del intervalo. Los resultados para los demás porcentajes se muestran en la quinta columna.

Tabla 1. Solución aproximada y exacta para $m=3,200$ para una $\lambda = 0.0035\%$

$\varepsilon(\%)$	Iteración máximo	Iteración optimo	$C(I)_{aprox}$	$C(I)$
6	4	5	2,202	$2,202 \leq C(I)_{opt} \leq 2,334$
4	62	1339	2,309	$2,309 \leq C(I)_{opt} \leq 2,334$
2	338	1341	2,309	$2,309 \leq C(I)_{opt} \leq 2,334$
1	1338	1343	2,309	$2,309 \leq C(I)_{opt} \leq 2,332$
0	1378	2655	2,329	$C(I)_{opt} = C(I)_{aprox} =$
(exacta)				2,329

En esa tabla 1 se observa que en todos los casos la solución exacta está dentro de los intervalos calculados de las soluciones aproximadas. El costo de obtener la

solución exacta se ve reflejada en el número de iteraciones requeridas para determinar el valor máximo y para determinar que este máximo es la solución óptima.

A causa de la diferencia tan grande de iteraciones, entre la solución exacta y la aproximada, para este problema ($m=3,200$) es posible calcular las soluciones enteras aproximadas para cualquier λ siempre y cuando $\varepsilon \geq 1\%$. Para calcular la solución óptima exacta de cualquier λ se necesita de cómputo paralelo por la gran cantidad de cálculos y memoria requeridos.

A causa de que el método del árbol de cubos es eficaz es posible aplicarlo en determinar la solución óptima de problemas de optimización mas complejos como la selección del subconjunto de objetos con un parámetro fijo, multirestricciones y multicriterial.

4.2.5 Algoritmo de búsqueda en el intervalo de estabilidad de la solución óptima con un parámetro fijo

En los párrafos siguientes enseñamos varias modificaciones del método de árbol de cubos que pueden ayudar a resolver problemas de optimización mas complejos, multirestricciones y multicriterial. Estas modificaciones se fundamentan en el método "aproximative-combinatorial" que se detalla en los artículos [9, 10].

Este algoritmo puede ser modificado para determinar, no sólo la solución óptima, también puede determinar la región de estabilidad de la solución óptima. A este subconjunto de soluciones la denotamos por $\Omega(R)$, donde:

$$\Omega(R) = \{\omega \in I \mid C(I, \omega_{opt}) - R \leq C(\omega) \leq C(I, \omega_{opt})\} \text{ donde } R \geq 0.$$

Entonces $\Omega(R)$ es el conjunto de todas las soluciones integrales no diferentes para la significación de la función a la solución óptima en mas que $R \geq 0$.

Para buscar $\Omega(R)$ es necesario modificar las Reglas para el rechazo de subárboles no óptimales del Método de árbol de Cubos. La modificación que se tiene que incluir en los bloques 3 y 4 es la siguiente:

Para todos los subárboles $A(l)$ para los cuales:

$$C(I, \tilde{\omega}) > C(L, \omega^0, l) + R$$

los excluimos del cálculo (bloque 3).

En el proceso del algoritmo, se almacena en la memoria todas las soluciones para las cuales (bloque 4):

$$C(I, \omega^0, l) \geq C(I, \omega_{opt}) - R$$

Después de realizar los cálculos se tendrá almacenada en la memoria el conjunto $\Omega(R)$.

Este conjunto se puede utilizar para buscar el óptimo de la tarea cuando se tiene mas de una restricción. Por ejemplo, se tienen k restricciones lineales y conocemos la solución $\omega_k \in I$ con el funcional $C(I, \omega_k)$ que satisface todas las restricciones, pero no puede ser el óptimo.

Es necesario buscar el óptimo $\omega_{opt} \in I$ con su $C(I, \omega_{opt})$ para la tarea con todas sus restricciones. Resolvemos n tareas presentadas en el punto 4.1, para cada una de las restricciones y buscamos para cada una de ellas las soluciones óptimas: $\omega_{opt}^1 \subset I$ con el funcional $C(I, \omega_{opt}^1)$, $\omega_{opt}^2 \subset I$ con el funcional $C(I, \omega_{opt}^2)$, ..., $\omega_{opt}^n \subset I$ con el funcional $C(I, \omega_{opt}^n)$. Posteriormente buscamos para cada tarea sus $R_j = C(I, \omega_{opt}^j) - C(I, \omega_0)$, claro que $R_j \geq 0$ para todos los $j = 1, 2, \dots, n$. Para cada tarea buscamos los conjuntos:

$$\Omega_j(R_j) = \{\omega \subset I \mid C(I, \omega_{opt}^j) - R_j \leq C(\omega) \leq C(I, \omega_{opt}^j)\}$$

Posteriormente buscamos la intersección

$$\Omega_0 = \bigcap_{j=1}^n \Omega_j(R_j)$$

$\Omega_0 \neq \emptyset$ porque siempre $\omega_b \in \Omega_j$, $j = 0, 1, 2, \dots, n$.

En correspondencia con el método combinatorio aproximacional [9], en este caso siempre $\omega_{opt} \in \Omega_0$, por eso

$$C(I, \omega_{opt}) = \max_{\omega \in \Omega_0} C(\omega)$$

Si tenemos n criterios lineales $C_1(\omega)$, $C_2(\omega)$, ..., $C_n(\omega)$, para cada uno de ellos buscamos las soluciones óptimas

$$C_1(I, \omega_{opt}^1), C_2(I, \omega_{opt}^2), \dots, C_n(I, \omega_{opt}^n)$$

Si tenemos una solución $\omega_b \subset I$ para el cual conocemos $C_1(\omega_b)$, $C_2(\omega_b)$, ..., $C_n(\omega_b)$ calculamos $R_j = C_j(\omega_{opt}^j) - C_j(\omega_b)$, buscamos:

$$\Omega_j(R_j) = \{\omega \mid C_j(I, \omega_{opt}^j) - R_j \leq C_j(\omega) \leq C_j(I, \omega_{opt}^j)\}$$

Posteriormente determinamos:

$$\Omega_0 = \bigcap_{j=1}^n \Omega_j(R_j)$$

$\Omega_0 \neq \emptyset$ porque $\omega_b \in \Omega_0$. Todos $\omega \in \Omega_0$ son soluciones que tienen $C_j(\omega)$ mejor ($\geq$) que solución $\omega_b \in \Omega_0$ para todos los criterios $C_j(\omega)$ al mismo tiempo, y peor ($\leq$) con la solución $\omega_{opt}^j \in \Omega$ no mas que $R_j \geq 0$.

Todos los $\omega \notin \Omega_0$ son peores que la solución ω_b en al menos un criterio.

5 Conclusiones

El método y el algoritmo del árbol de cubos son eficaces para resolver problemas de optimización discreta. En la gran cantidad de problemas resueltos se determinó la solución exacta sin requerir un número elevado de iteraciones, en el peor de los casos fue $O(m^4)$.

Las reglas de rechazo en las cuales se fundamenta este método, permiten que se puedan modificar para resolver otro tipo de problemas, como se mostró en la determinación del intervalo de estabilidad de la solución óptima. Con lo cual se muestra que el método del árbol de cubos es flexible para resolver un gran número de problemas de optimización discreta.

El método del árbol de cubos, fundamentado matemáticamente, es una buena opción para resolver problemas que se venían resolviendo con otros métodos exactos, pero con la diferencia de que nuestro método se puede aplicar a solucionar problemas de optimización discreta de dimensiones mayores. Además, en el caso de que el problema sea lo suficientemente grande y requiera de una gran cantidad de memoria y procesamiento, que no sea posible satisfacerla con una computadora de un solo procesador, es posible utilizar el método con una variante para determinar la solución aproximada, donde la solución óptima estará en el intervalo dado por la precisión requerida.

Como se mostró en el procedimiento del método árbol de cubos, las reglas de rechazo nos indican que en varios subárboles se podría encontrar la solución óptima, donde cada subárbol es un conjunto disjunto. Por tal motivo es posible calcular cada uno en forma independiente, lo cual se puede aprovechar para utilizar las computadoras paralelas, donde cada procesador calcule un subárbol. Nuestros primeros análisis y resultados, nos indican que para una máquina paralela con 32 procesadores es posible incrementar el número de variables de 3 a 4 veces más que la versión secuencial.

Referencias

1. Birkhoff G., "Lattice Theory", second edition Amer., Math Soc., Providence, 1948
2. Grätzer G. "General Lattice Theory" Akademie-Verlag Berlin, 1978.
3. Cook W. J., Cunningham W. H., Pulleyblank W. R., Schrijver A. "Combinatorial Optimization" A Wiley-Interscience Publication, John Wiley and Sons Inc, New York 1998.
4. Nemhauser G. L., Wolsey L. A. "Integer and Combinatorial Optimization" A Wiley-Interscience Publication, John Wiley and Sons Inc., New York 1999
5. Khachaturov V. R. "The modeling and methods of solutions of multiextremal allocations problems and the use of supermodular function properties in the Boolean lattices", Algorithms and Languages of algorithms, Moscu: "Nauka", 1986
6. Khachaturov V. R. "Multiextremal allocation problems (models and solution methods)". Studia Scientiarum Mathematicarum Hungarica 27, 1992.

7. Khachaturov et others. "Combinatorial methods and algorithms for solving problems of discrete optimization with large dimensionality" Moscu: "Nauka", 2000.
8. Pisinger D., "An exact algorithm for large multiple knapsack problems", *European Journal of Operational Research*, 114 528-541 (1999)
9. Khachaturov V. R. "The Approximative-Combinatorial Meted and Some of its Applications", Moscow, Computational Mathematics and Mathematical Phisics, Vol 14, 1974.
10. Khachaturov V. R. "The Approximative-Combinatorial Meted for Decomposition and Composition of Sistems and Finite Topology Spaces, Lattices and Optimisation", Moscow, Computational Mathematics and Mathematical Phisics, 1985.

Semejanza entre objetos tridimensionales aplicando redes neuronales Fuzzy ART

A. Angeles-Yreta, J. Figueroa-Nazuno

Centro de Investigación en Computación, Instituto Politécnico Nacional,
Unidad Profesional "Adolfo López Mateos" -Zacatenco- 07738 D.F, México
malberto@sagitario.cic.ipn.mx, jfn@cic.ipn.mx

Resumen. El problema de definir y computar semejanza entre objetos ha sido de gran interés por su aplicabilidad, estos objetos pueden ser series de tiempo, imágenes, objetos tridimensionales, etc., es así, un problema lejos de ser trivial. Específicamente este trabajo se enfoca en el problema de semejanza entre figuras geométricas tridimensionales creadas por un sistema típico de Diseño Asistido por Computadora (DAC). Múltiples técnicas han sido propuestas para determinar semejanza entre objetos tridimensionales; la mayoría de ellas contempla al menos una fase de extracción de características antes de determinar cualquier relación de semejanza entre dichos objetos. Resultados experimentales muestran la aplicabilidad de Redes Neuronales Artificiales (RNA) Fuzzy ART para el cálculo de semejanza y por ende clasificación de objetos tridimensionales excluyendo cualquier etapa de extracción de características. En la etapa final se logra la unificación de las clases obtenidas por las RNA utilizando el concepto de distancia de Hamming.

1 Introducción

El problema de determinar semejanza entre objetos es en esencia la tarea medular de muchas aplicaciones de Minería de Datos [1][11], tales como: Descubrimiento de Reglas, Búsqueda por Contenido y por supuesto Clasificación.

Algunos métodos de búsqueda de semejanza entre objetos tridimensionales utilizan técnicas de Extracción de Características [2][3][4], ello implica algún tipo de transformación del objeto tridimensional a un vector de características, con el objetivo de preservar, descubrir o seleccionar alguna propiedad, de esta forma ese vector de características puede ser manejado como una serie de tiempo que se define como una secuencia de observaciones realizadas a un Δt constante [1][5][7], otros métodos consideran los objetos tridimensionales como una secuencia de imágenes permitiendo aplicar métodos de búsqueda de semejanza entre imágenes [8][10], también se encuentran métodos basados en histogramas, que aunque puede ser un caso particular de los métodos basados en Ex-tracción de Características, usualmente es definido como una clase diferente. Por último existen métodos híbridos, que pretenden tomar características de distintos métodos de acuerdo a los objetos tridimensionales en cuestión. En este trabajo se presenta un método de búsqueda de

semejanza entre objetos tridimensionales sin ninguna etapa de Extracción de Características.

El presente trabajo está organizado de la siguiente forma. En la sección 2 se describe la RNA Fuzzy ART utilizada para la clasificación de series de tiempo y de la cual serán requeridas tres instancias para clasificar los vectores que componen a las figuras geométricas tridimensionales. En la sección 3 se describe la codificación de los objetos tridimensionales empleada para este trabajo, en la sección 4 se describe la abstracción de los Objetos Tridimensionales como Series de Tiempo; en la sección 5 se define la Metodología para las pruebas experimentales, Resultados en la sección 6 son presentados y finalmente en la sección 7 las Conclusiones.

2 RNA Fuzzy ART.

El cómo aprendemos cosas sin olvidar las ya aprendidas inspiró a Grossberg y Carpenter [6] a desarrollar una teoría, Adaptive Resonant Theory (ART), de ésta se desprenden una serie de arquitecturas RNA del tipo auto-organizadas. ART establece que la Memoria de Largo Plazo (LTM Long Term Memory) solamente registra un *patrón* cuando difiere de los demás en relación a un *umbral*, y que actualiza información en dicha memoria cuando se *entra* en un estado de *resonancia* debido a la semejanza de dicho patrón de entrada con los ya almacenados.

ART, ART2, FUZZY-ART, ARTMAP son parte de la familia de arquitecturas ART, con algunas modificaciones éstas permiten clasificar vectores de entrada binarios y análogos, así como el tipo de entrenamiento (supervisado y no supervisado). De aquí en adelante nos referiremos a la RNA Fuzzy ART. En la figura 1 se muestra la arquitectura de la RNA Fuzzy ART.

La RNA Fuzzy ART clasifica vectores de entrada al asignarles una clase dentro de las N clases posibles. La semejanza entre vectores de entrada esta determinada por tres parámetros $\alpha > 0$, ρ y β con rango $[0, 1]$.

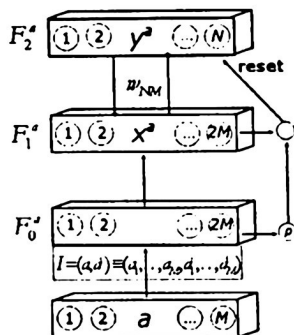


Fig. 1. El vector de entrada a , donde cada $a_i \in [0, 1]$ es el patrón a clasificar, para evitar una proliferación de clases el vector de entrada a de longitud M es convertido a un vector I de longitud $2M$ en la capa F_0^a , W_{NM} es la matriz de pesos, en ella se encuentran los vectores prototipo, F_2^a es una capa adaptativa o de contención.

A continuación se muestra el algoritmo de la RNA Fuzzy ART.

1. Establecer parámetros: $\alpha > 0$, $\beta \in [0,1]$, y $\rho \in [0,1]$
2. Iniciación de matriz de pesos $w_{11} = \dots = w_{NM} = 1$
3. Mientras existan vectores de entrada
 - 3.1. Establecer el vector de entrada $\mathbf{a}$,
 $I = (a, a^c) \equiv (a_1, \dots, a_M, a_1^c, \dots, a_M^c)$, donde $a_i^c \equiv 1 - a_i$
 - 3.2. Mientras no sea encontrada la categoría del vector de entrada

$$T_j(I) = \frac{|I \wedge w_j|}{\alpha + |w_j|}$$
 - 3.2.1. Calcular T_j para cada nodo "clase",
 - 3.2.2. Encontrar el nodo "ganador", $T_j = \max\{T_j : j = 1, \dots, N\}$
 - 3.2.3. Si $\frac{|I \wedge w_j|}{|I|} \geq \rho$ es VERDADERO $\equiv$ SIMILAR, entonces
 actualizar pesos $w_j^{(new)} = \beta(I \wedge w_j^{(old)}) + (1 - \beta)w_j^{(old)}$
 De otra forma, deshabilitar al nodo "clase" j e ir al paso 3.2.
 - 3.3. Probar si el vector de entrada $\mathbf{a}$ ha encontrado su categoría
4. Probar condición de paro

3 Codificación de objetos tridimensionales

Aunque existen varias codificaciones de objetos tridimensionales, en este trabajo usamos una de las más sencillas, el formato *Object File Format* (OFF), ésta consiste en la definición de un arreglo de vértices V (1); este arreglo contiene todos los vértices que componen al objeto tridimensional, cada vértice v_i (2) es una tripleta con tres escalares que representan a los puntos en el plano x, y y z .

$$V = \{v_1, v_2, v_3, \dots, v_k\} \quad (1)$$

$$v_i = \{x_i, y_i, z_i\} \quad (2)$$

Adicionalmente se define un arreglo de vértices F (3) que indica la secuencia en que estos vértices deben ser dibujados, este arreglo se conoce como índice de caras.

$$F = \{v_{c1}, v_{c2}, v_{c3}, \dots, v_{cn}\}, \text{ donde } 1 \leq c_i \leq k \quad (3)$$

Con los arreglos V y F , construimos una matriz de 3 por $(c \cdot k)$ que es un arreglo con los vértices en la secuencia en que éstos deben ser dibujados. A continuación se muestra el pseudo-código utilizado para la construcción de dicha matriz:

```
function [M] = BuildMatrix(Vertices, Faces)
    for i=1:length(Faces)
        M(i) = GetVertex(Faces(i), Vertices);
    end
    return M;
end function
```

De lo anterior se formula que:

$$M = \{V_{F_1}, V_{F_2}, V_{F_3}, \dots, V_{F_c}\} \quad (4)$$

4 Objetos tridimensionales como series de tiempo

Una vez que logramos tener un arreglo M de acuerdo a (4) con los vértices en la secuencia en que éstos deben ser dibujados, creamos tres series de tiempo X , Y y Z , donde los elementos de la serie de tiempo X , son los componentes del plano x de los vértices, de la misma forma son constituidas la serie de tiempo Y y Z . En la figura 2 se muestra un ejemplo en el que el objeto tridimensional es representado por tres series de tiempo

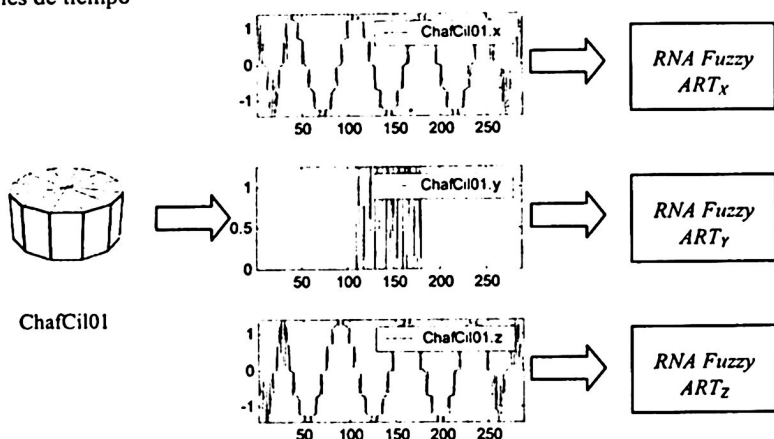


Fig. 2. En esta figura se muestra cómo un objeto tridimensional puede ser abstraído en tres series de tiempo, tomando para la serie de tiempo X , la columna 1 (plano x), de la matriz M , y así sucesivamente con las series de tiempo Y y Z .

La creación de tres RNA Fuzzy ART que comparten los mismos valores para los parámetros rho, alpha y beta es necesaria para clasificar el objeto tridimensional en sus diferentes componentes. Ello implica una etapa final para la *unificación* de clases obtenidas por las RNA Fuzzy ART. En la tabla 1 se muestran resultados parciales de un caso particular.

Las tres RNA Fuzzy ART comparten los mismos valores para los parámetros rho, alpha y beta, estas RNA generarán tres clasificaciones que pueden ser distintas para cada objeto tridimensional de un conjunto dado.

Una vez obtenidas las tres listas de *clases*, las cuales contienen las clasificaciones de los objetos tridimensionales de acuerdo a su semejanza, se convierten los índices de clases a cadenas binarias concatenadas para calcular su distancia de Hamming con

respecto de todas las demás y así *unificar* las clases. La Tabla 2 muestra un ejemplo de conversión a cadena binaria de los índices de clases y su distancia de Hamming.

Tabla 1. En la columna de la extrema derecha se encuentran los objetos tridimensionales que han sido ingresados a las tres RNA en forma de series de tiempo y su respectiva clase en el plano *X*, *Y* y *Z*.

	Clase en X	Clase en Y	Clase en Z
ChafCil01	0	0	0
ChafCil02	0	0	0
ChafCil03	0	0	0
ChafCil04	0	0	0
ChafCil05	1	0	1
ChafCil06	2	1	2
Ext_C01	3	1	3
Ext_C02	3	1	3
Ext_C03	3	1	3
Ext_C04	3	1	3

Tabla 2. El primer objeto de la lista que resulta ser ChafCil01 calcula la distancia de Hamming de sus clases en *X*, *Y* y *Z* con el resto de los objetos, la primer clase se creará para todos aquellos que su distancia de Hamming es menor o igual a 1.

	ChafCil01:000000000000000000000000
ChafCil01:000000000000000000000000	0
ChafCil02:000000000000000000000000	0
ChafCil03:000000000000000000000000	0
ChafCil04:000000000000000000000000	0
ChafCil05:000000010000000000000000	2

En la tabla 2 se muestra un ejemplo en el que del objeto ChafCil01 al ChafCil04 comprenden una clase, digamos la 0, debido a que su distancia de Hamming es menor o igual a 1, es decir, el objeto ChafCil05 será ahora el nuevo candidato a computar su distancia con resto de los objetos para crear otra clase, digamos 1, sin considerar los que ya pertenecen a una clase, que son para este caso los objetos de la clase 0.

5 Metodología

En resumen la abstracción de objetos tridimensionales a series de tiempo nos permite el uso de técnicas probadas para la búsqueda de semejanza entre series de tiempo

como es la RNA Fuzzy ART. A continuación se muestra la metodología para determinar semejanza entre objetos tridimensionales.

- I. Por cada objeto tridimensional codificado de acuerdo a la sección 3.
- II. Crear tres series de tiempo $X_{Objeto1}$, $Y_{Objeto1}$ y $Z_{Objeto1}$.
- III. Redimensionar todas las series de tiempo a una longitud constante. En este trabajo utilizamos 1000 puntos por cada serie de tiempo.
- IV. Ingresar las series de tiempo $X_{Objeto1}$, $Y_{Objeto1}$ y $Z_{Objeto1}$ a las RNA Fuzzy ART_x, RNA Fuzzy ART_y, RNA Fuzzy ART_z, respectivamente, las tres RNA comparten los mismos valores para rho, alpha y beta, así como el mismo número de clases de salida.
- V. Unificar las clases con la distancia de Hamming, de acuerdo a la sección 4.
- VI. Listar los objetos de acuerdo a su clase.

6 Resultados

El experimento consist en 46 objetos tridimensionales, el objetivo era poder clasificarlos de acuerdo a su semejanza; a continuación se muestran las 18 clases que se obtuvieron con los siguientes parámetros:

`net.InitializeFuzzyART(0.7, 0.1, 1); //Rho, Alpha y Beta`

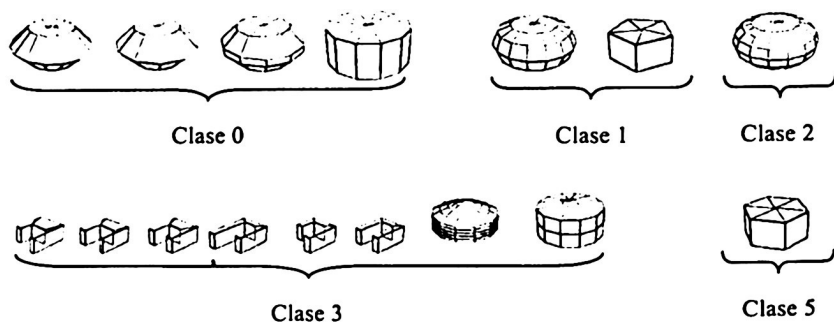


Fig. 3. En esta figura se muestran 5 clases, la clase 3 los últimos dos objetos parecen un tanto distintos a los 6 primeros, esta clasificación es correcta, el proceso de redimensionar las series de tiempo a una longitud constante obliga a la RNA Fuzzy ART a colocarlos en la misma clase.

El método propuesto genera clases bien definidas en base a su semejanza, con los parámetros correctos la clasificación de objetos tridimensionales es muy factible. El problema que presenta la clase 3, y quizá la clase 12 se deben al método para redimensionar las series de tiempo. Este trabajo no se enfoca en la técnica para redimensionar las series de tiempo, la repercusión de redimensionar las series de tiempo a una longitud constante es tema para un trabajo futuro.

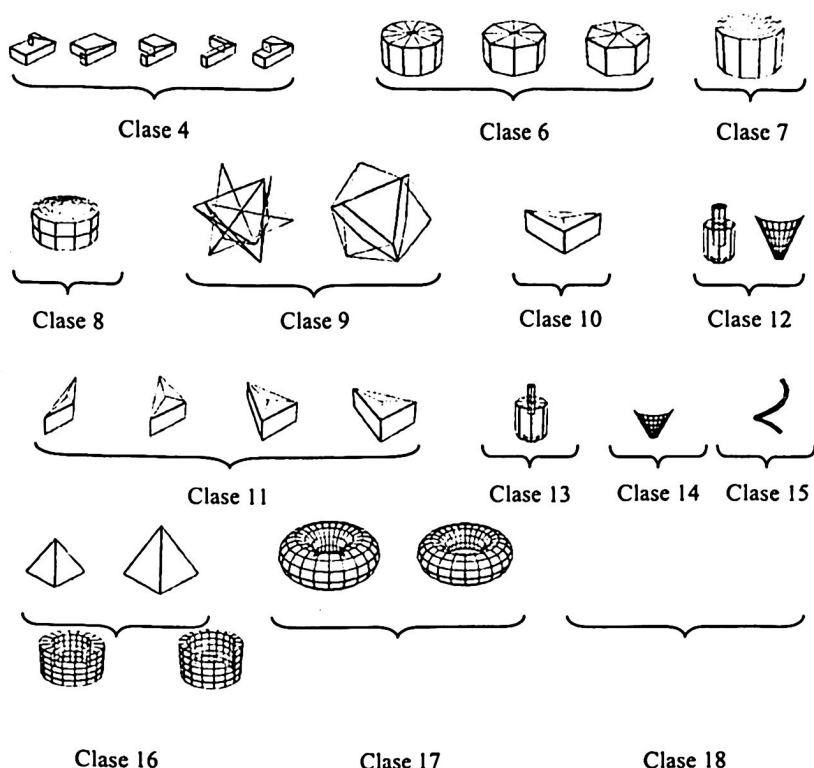


Fig. 4. En esta figura podemos observar el resto de las clases creadas con el método propuesto.

7 Conclusiones

Resultados experimentales de este trabajo muestran la viabilidad de usar RNA Fuzzy ART y la distancia de Hamming para la búsqueda de semejanza en objetos tridimensionales. Siendo así uno de los métodos más sencillos y directos para dicha tarea. A diferencia de otros métodos, la combinación de RNA Fuzzy ART y distancia de Hamming resulta única; trabajos futuros en base a éste resultan alentadores, algunos de ellos son: estudiar más a fondo el proceso para redimensionar las series de tiempo y su repercusión sobre la clasificación, así como distintas formas de combinar las series de tiempo para representar los objetos tridimensionales a clasificar.

Referencias

1. Angeles-Yreta, A., Solís-Estrella, H. Landassuri-Moreno, V. Figueroa-Nazuno, J.: Similarity Search In Seismological Signals. Fifth Mexican Internacional Conference on Computer Science. Colima, México. (September 2004) 50-56.
2. Hlavaty, T., Skala, V.: Survey of Methods for 3D Model Feature Extraction. Bulletin of IV. Seminar "Geometry and Graphics in Teaching Contemporary Engineer". Szczyrk, Poland. No: 13/03. (2003) 5-8.
3. Popivanov, I., Miller, R.: Similarity Search Over Time-Series Data Using Wavelets. Proceedings of the 18th International Conference on Data Engineering. San José, California. (February 2002) 212-224.
4. Jianguo, L. Jingdong, W.: Fuzzy ART based Texture Classification. Report of the Neural Networks Course, Department of Automation, TsingHua University. Beijing, China, January 2002.
5. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Knowledge and Information Systems. (August 2000) 263-286.
6. Carpenter, G., Grossberg, S., Rosen, D.: Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System. Neural Networks. Pergamon Press. Vol. 4, pp. 759-771.
7. Kriegel H.-P., Brecheisen S., Krger P., Pfeifle M., Schubert M.: Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects. Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'03). San Diego, California. (2003).
8. Bradski, G., Grossberg, S.: Fast-learning VIEWNET architectures for recognizing 3-D objects from multiple 2-D views. Neural Networks. Vol. 8. No. 7-8. (1995) pp. 1053-1080.
9. Feature Extraction, Construction and Selection: A Data Mining Perspective. Kluwer Academic Publishers. Edited by Huan Liu, Hiroshi Motoda. Second ed. (1998).
10. Lankalapalli K., Chatterjee S, Chang T.C.: Feature Recognition Using ART2: a Self-organising Neural Network. Journal of Intelligent Manufacturing. Vol. 8. No.3. (1997) pp. 203-214.
11. Zaharia T., Preteux F.J. 3D Shape-based retrieval within the MPEG-7 framework. Nonlinear Image Processing and Pattern Analysis XII (Proceedings of SPIE Volume 4304). San Jose, California. (2001)pp. 133-145.

Software Technologies and Data Bases

Tecnologías de software y bases de datos

UML Model of the IEEE 802.11 Privacy Service using AES-CCM (Advanced Encryption Standard, Counter-Mode/CBC-MAC) and AES Implementation

Sugehi Marina Merino Higareda, Miguel Ángel León Chávez

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación,
14 Sur y Av. San Claudio, C.P. 72570, Puebla, México.
msugehi@hotmail.com, mleon@cs.buap.mx

Abstract. Nowadays, mobility plays an important role in our society. Thanks to the wireless networks a real mobility in the devices has been obtained. The IEEE 802.11 standard [1] is one of the most deployed wireless technologies all over the world. Its main characteristics are simplicity, flexibility and cost effectiveness. However, the communication medium is insecure and in some applications the insecurity is an unwanted characteristic. This standard provides users with three security services: Authentication, Privacy, and Integrity. The privacy service is implemented by WEP (Wired Equivalent Privacy) protocol, which has demonstrated to be vulnerable [2, 3, 4]. This paper discusses the WEP vulnerabilities, and proposes to replace it by AES-CCM (Advanced Encryption Standard, Counter-Mode/CBC-MAC). The Unified Modeling Language (UML) model of the IEEE 802.11 privacy service using AES-CCM and AES implementation are presented.

1 Introduction

The purpose of the IEEE 802.11 standard [1] is to provide wireless connectivity to automatic machinery, equipment, or stations that require rapid deployment, which may be portable or hand-held, or which may be mounted on moving vehicles within a local area.

The IEEE 802.11 is part of a family of standards for local area networks (LANs), this standard defines two OSI layers only: Physical (PHY) layer and Data Link layer (DLL), this last one is divided in Logical Link Control (LLC) sub-layer and Medium Access Control (MAC) sub-layer. At the PHY layer, the standard defines the protocols and compatible interconnections of communication equipment via the air, radio or infrared like transmission medium. The MAC sub-layer defines two medium access coordination functions, one centralized (Point Coordination Function, PCF) and other distributed (Distributed Coordination Function, DCF) which implements Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol and Request To Send and Clear To Send (RTS/CTS) protocol, and should be implemented obligatorily in all the stations.

The MAC sub-layer provides users with the following services: MSDU delivery, Authentication, Deauthentication, Association, Disassociation, Reassociation, Distribution, Integration, and Privacy.

The IEEE 802.11 defines three security services, as follows: Authentication, Privacy and Integrity. The privacy service is implemented by the WEP (Wired Equivalent Privacy) protocol, which has demonstrated to be vulnerable [2, 3, 4].

This paper proposes to replace WEP protocol by AES-CCM (Advanced Encryption Standard, Counter-Mode/CBC-MAC). AES is based on the Rijndael algorithm, which is a powerful encryption and efficient algorithm. AES is a block cipher, with both keys and blocks of 128, 192, or 256 bits long.

Why do you use AES? Because AES is a cipher with a simple and elegant structure, and the key scheduling algorithm [5] shows a satisfactory security margin against cryptanalytic attacks.

AES, as the old standard DES, works in several modes. This paper uses the CCM (Counter-Mode/CBC-MAC) mode [6], because it provides authentication and encryption using a single block cipher key that is established beforehand. Thus, CCM requires a well-designed key management structure.

In order to implement the IEEE 802.11 privacy service using AES-CCM this paper presents the UML model of MAC sub-layer in DCF mode, which includes the privacy service.

The rest of this paper is organized as follows: Section 2 discusses the IEEE 802.11 security services, section 3 presents AES-CCM. Section 4 presents the UML model, which includes the analysis and design models, of the privacy service using AES-CCM. Section 5 presents the AES implementation. Section 6 presents the conclusions and our future work.

2 IEEE 802.11 Security Services

Three main security services are defined by the IEEE 802.11, as follows: Authentication, Integrity and Privacy. The *authentication* service verifies the supposed identity of a user or a system. The *integrity* service protects the data against non-authorized modifications, insertions or deletions. The *privacy* service protects the data against non-authorized revelations.

The authentication service is defined in two subtypes: Open System and Shared Key. In the former, no any authentication mechanism is used by the stations. In the latter, a shared secret key is used by the stations.

The integrity service is implemented by the Cyclic Redundancy Check (CRC)-32 algorithm, but the checksum that it creates is a noncryptographic value, known attacks, such as side-channel attacks, can compromise the data's integrity.

The privacy service is implemented by the WEP protocol, which is based on the RC4 symmetric algorithm. RC4 is a quite powerful crypto algorithm. Nevertheless, WEP takes a poor approach for using it.

Several vulnerabilities of the WEP protocol were discovered and analyzed in the last few years [2, 3, 4].

WEP uses a 64 key bit long, formed by an Initialization Vector (IV, 24 bits) and a secret key (typically, 40 bits long).

One of WEP's biggest downfalls is that its secret key is relatively shorter than other security protocols' key

Other problem is the IV because it is sent to the receiver in plaintext, which means that attackers can see the first 24 bits of every key used by WEP. Furthermore, the fact that the IV is so short nearly guarantees that it will be used for multiple messages. In fact, the same IV might be reused in as little as half of a day if there is significant activity over a company's WLAN. An attacker could easily collect an IV and use it to retrieve the key that the Access Point (AP) and wireless devices use.

WEP security also suffers from a poor solution for key management, which can leave the keys in a device unchanged for long periods of time. If the device were lost or stolen, an attacker could use the key to compromise not only that device but any other devices sharing the same key.

This is way this paper proposes to change the WEP protocol by the AES-CCM algorithm in order to implement the privacy service.

3 AES-CCM

Rijndael is a block cipher designed by J. Daemen and V. Rijmen and has been adopted as the Advanced Encryption Standard (AES). AES [5] is designed for use with blocks and keys of 128, 192 and 256 bits long. For simplicity, we use blocks and keys of 128 bits long.

AES consists of 10 rounds. Each round has a round key, derived from the original key. The original key is used in round zero. A round initiates with an input of 128 bits and produces an output of 128 bits. The algorithm treats 128-bit input block as a group of 16 bytes organized in a 4×4 matrix called State matrix. There are four basic steps, named layers, which are used to form the rounds:

1. The ByteSub (BS) Transformation: This nonlinear layer is for resistance differential and linear cryptanalysis attacks.
2. The ShiftRow (SR) Transformation: This linear mixing step causes diffusion of the bits over multiple rounds.
3. The MixColumn (MC) Transformation: This layer has a purpose similar ShiftRow.
4. AddRoundKey (ARK): The round key is or-exclusived with the result of the above layer.

Putting everything together, we obtain the Rijndael Encryption:

1. ARK, using the 0th round key.
2. Nine rounds of BS, SR, MC, ARK, using round keys 1 to 9.
3. A final round: BS, SR, ARK, using the 10th round key.

A round key is derived from the original key through a process called Key Scheduling. The Rijndael decryption algorithm operates similarly by applying inverse of all transformations described above in reverse order [5, 7].

Next we shall briefly describe the four AES transformations and the CCM mode.

3.1 The ByteSub Transformation

Each input byte of the State matrix is independently replaced by another byte from a look-up table called S-box. The S-box is a 256-entry table composed of two transformations: First each input byte is replaced with its multiplicative inverse in $GF(2^8)$ with the element $\{00\}$ being mapped onto itself; followed by an affine transformation over $GF(2^8)$ [5, 7]. For decryption, inverse S-box is obtained by applying inverse affine transformation followed by multiplicative inversion in $GF(2^8)$ [5].

3.2 The ShiftRow Transformation

It is a cyclic shift operation where each row is rotated cyclically to the left using 0, 1, 2 and 3-byte offset for encryption while for decryption, rotation is applied to the right.

3.3 The MixColumn Transformation

In this transformation, the columns of the State matrix are considered as polynomials over $GF(2^8)$ and multiplied modulo x^4+1 with a fixed polynomial $c(x)$, given by:

$$c(x) = 3x^3 + x^2 + x + 2. \quad (1)$$

This polynomial is coprime to x^4+1 and therefore invertible. This can be written as a matrix multiplication.

$$b(x) = c(x) \text{ XOR } a(x). \quad (2)$$

Similarly, for decryption process, we compute Inverse MixColumn, by multiplying each column of State matrix by a constant fixed matrix.

3.4 The RoundKey Addition

The output of MC is XOR-ed with corresponding round sub-key derived from user key. The ARK step is essentially same for encryption and decryption processes.

3.5 The Key Schedule

The original key consists of 128 bits, which are arranged into a 4x4 matrix of bytes. This matrix is expanded by adjoining 40 more columns, as follows. Label the first four columns $W(0)$, $W(1)$, $W(2)$, $W(3)$. The new columns are generated recursively. Suppose columns up through $W(i-1)$ have been defined. If i is not a multiple of 4, then:

$$W(i) = W(i-4) \text{ XOR } W(i-1). \quad (3)$$

If i is a multiple of 4, then:

$$W(i) = W(i-4) \text{ XOR } T(W(i-1)). \quad (4)$$

Where $T(W(i-1))$ is the transformation of $W(i-1)$ obtained as follows. Let the elements of the column $W(i-1)$ be a, b, c, d . Shift these cyclically to obtain b, c, d, a . Now replace each of these bytes with the corresponding element in the S-box from the ByteSub step, to get 4 bytes e, f, g, h . finally, compute the round constant:

$$r(i) = 00000010^{(i-4)/4}. \quad (5)$$

In $GF(2^8)$ (recall that we are in the case where i is a multiple of 4). Then $T(W(i-1))$ is the column vector:

$$(e \text{ XOR } r(i), f, g, h) \dots \quad (6)$$

In this way, columns $W(4), \dots, W(43)$ are generated from the initial four columns. The round key for the i th round consists of the columns:

$$W(4i), W(4i+1), W(4i+2), W(4i+3). \quad (7)$$

3.6 CCM Mode

The CCM mode is designed to use the AES block cipher, to provide authentication and encryption using a single block cipher key that is established beforehand. CCM is intended for use in a packet environment; the plaintext input includes a header, which is authenticated but not encrypted, and a payload, which is authenticated and encrypted. CCM operates on the whole packets; it does not support partial processing or stream processing. Each packet must be assigned a unique value, called a nonce. The size of the nonce determines the maximum number of packets that can be authenticated and encrypted with a single block cipher key.

CCM processing expands the packet size by appending an encrypted authentication tag. Successful verification of the authentication tag provides assurance that the packet originated from a source with access to the block cipher key. Consequently, successful verification of the authentication tag also provides assurance that the packet was not altered after the generation of the authentication tag. Failed verification of the authentication tag is designed to reveal intentional, unauthorized modifications of the packet, as well as accidental modifications.

4 UML Model of the IEEE 802.11 Privacy Service

This section presents the UML model of the IEEE 802.11, in DCF mode, privacy service using AES-CCM.

The UML model is composed of the analysis and design models. The analysis model is composed by the use cases diagram and the class diagram. The design model is composed by the refined class, interaction, sequence, and state diagrams.

There are two categories of MAC services: the Station Services (SS) and the Distribution System Services (DSS). SS are as follows: Privacy, Authentication, Deauthentication, MSDU delivery. DSS are as follows: Reassociation, Disassociation, Association, Integration, and Distribution.

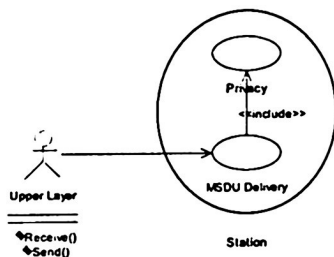


Fig. 1. Upper Layer Use Cases Diagram.. This shows an actor, who is an upper layer, which requests the SS; in this case only MSDU Delivery because the rest of the services are transparent for it. Note that MSDU Delivery includes the Privacy Service.

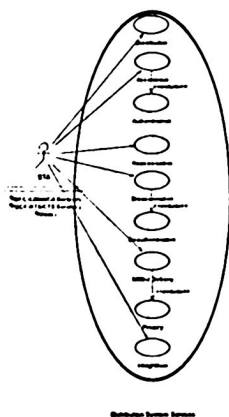


Fig. 2. Access Point Use Cases Diagram. This shows an actor, who is a Station (STA), which requests the DSS provided by an Access Point (AP).

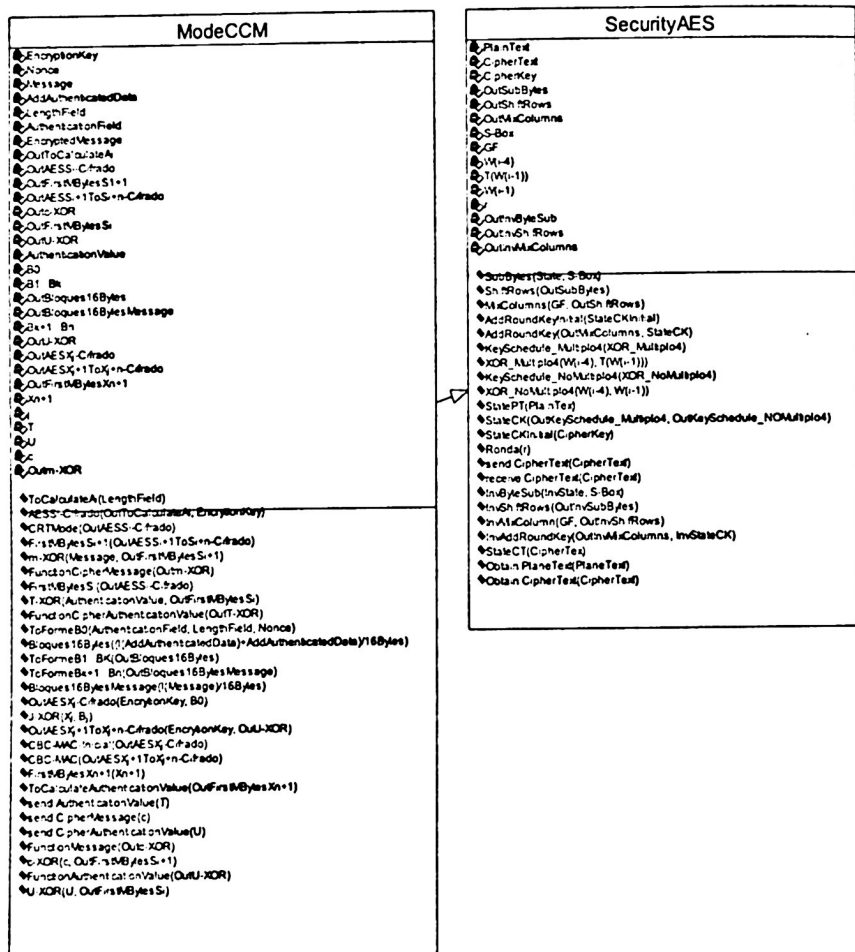


Fig. 4. AES-CCM Class Diagram. This shows the class diagram using AES-CCM, the classes are the following; CCMMode: It implements the CCM mode with an input block and key size of 128-bits. This class calls several methods of the PrivacyAES class for en/de-cryption. AESPrivacy: It implements the Rijndael algorithm, including the four basic layers and Key Scheduling process.

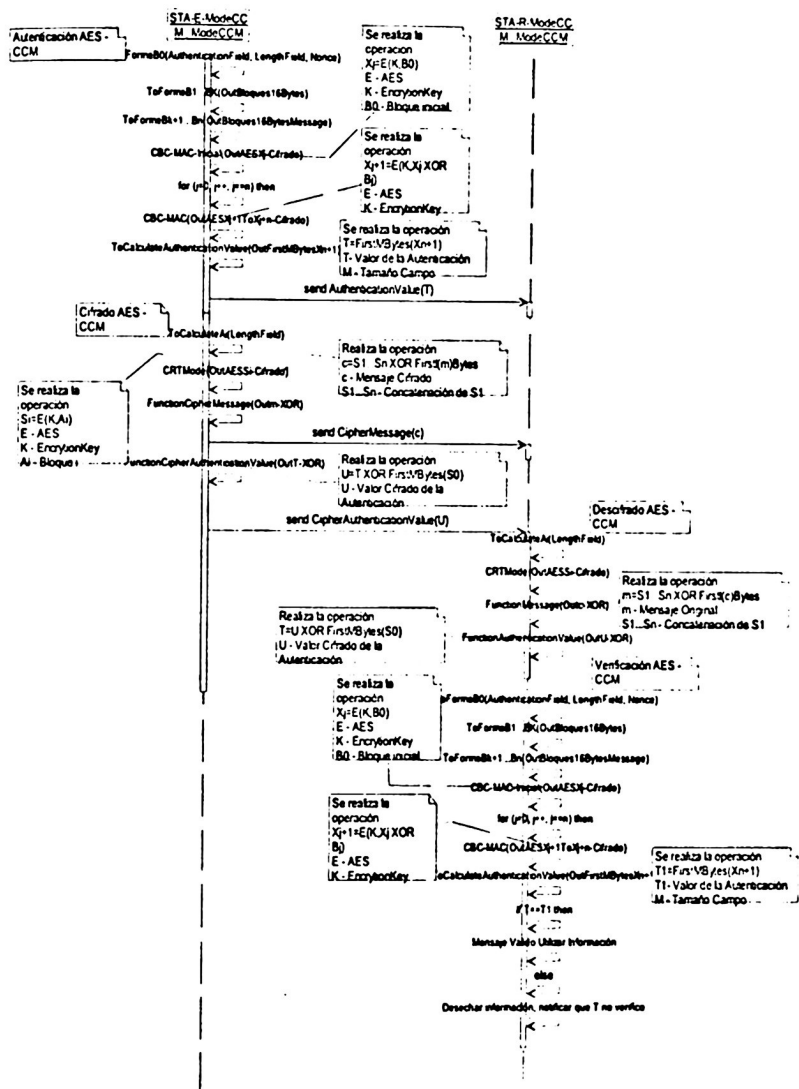


Fig. 5. AES-CCM Sequence Diagram. This shows the sequence diagram of the AES-CCM process, assuming that each STA has its secret key.

5 AES Implementation

The first step to implement AES-CCM is the AES implementation. This section presents the AES implementation is designed for use with blocks and keys of 128 bits long. This implementation used a single block cipher key that is established beforehand.

AES Program uses Microsoft Visual C++, Version 6 with platform Windows XP. This program is formed by two source files and two header files.

The MAIN.cpp source file is the main program, where the algorithm treats 128-bit input block (message) as a group of 16 bytes organized in a 4×4 matrix called State matrix and defined the key as a group of 16 bytes organized in a 4×4 matrix called CipherKey matrix.

The AES.cpp source file, it implements the Rijndael en/de-cryption algorithm, including the four basic layers and Key Scheduling process.

The AES.h header file is formed by the AES class, as follows:

```
class AES{
public:
    void ToObtainSetKey(unsigned char*Ckey);
    void CipherRijndael(unsigned char *State);
    void DecipherRijndael(unsigned char *State);
    void State(unsigned char *State);
    void RoundKeys();
private:
    unsigned char ExpKey[4][44];
    void KeyExpansion(unsigned char*Ckey);
    void ByteSub(unsigned char *State);
    void ShiftRow(unsigned char *State);
    void MixColumn(unsigned char* State);
    void AddRoundKey(unsigned char* State,int
Round);
    void InvByteSub(unsigned char *State);
    void InvShiftRow(unsigned char *State);
    void InvMixColumn(unsigned char* State);
    unsigned char Mult(unsigned char Byte, un-
signed char Val);
};
```

The AESTables.h header file, it is included S-Box, this is needed by the ByteSub transformation. Also it contains Inverse S-Box, this is used by the InvByteSub transformation.

6 Conclusions

This paper has presented the UML model of the IEEE 802.11, in DCF mode, privacy service using AES-CCM. The model includes the analysis and design models. We have discussed the IEEE 802.11 security services and the WEP vulnerabilities. For these reasons this paper has proposed to replace WEP protocol by AES-CCM.

This paper has presented the AES implementation. The implementation is designed for use with blocks and keys of 128 bits long. AES is a cipher with a simple and elegant structure, and the key scheduling algorithm shows a satisfactory security margin against cryptanalytic attacks.

This implementation is the first step to implement AES-CCM. To complete the implementation model is our future technology work as well as the performance analysis of AES-CCM into the IEEE 802.11g.

This paper uses the CCM (Counter-Mode/CBC-MAC) mode, because it provides authentication and encryption using a single block cipher key that is established beforehand. Thus, CCM requires a well-designed key management structure.

References

1. ANSI/IEEE Std 802.11, 1999 Edition. IEEE 802 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, (1999) 10-84.
2. J. R. Walker, Intel Corporation, Unsafe at any key size; An analysis of the WEP encapsulation, IEEE 802.11-00/362, October (2000) 1-9.
3. A. Stubblefield, J. Ioannidis, A. D. Rubin, Using the Fluhrer, Mantin, and Shamir Attack to Break WEP, Computer Science Dept. Rice University and AT&T Labs - Research, Florham Park, NJ, (1999) 1-11.
4. S. Fluhrer, I. Mantin and A. Shamir, Weaknesses in the Key Scheduling Algorithm of RC4, In 8th Annual Workshop on Selected Areas in Cryptography, August (2001) 1-23.
5. W. Stallings, Cryptography & Network Security: Principles and Practice, 2nd Edition, Upper Saddle River, NJ: Prentice Hall, (1998) 72-74, 121-130, 215-218.
6. D. Whiting, R. Housley and N. Ferguson, Counter with CBC-MAC (CCM)-AES Mode of Operation, NIST May (2004) 1-9.
7. J. Daemen, and V. Rijmen, The Design of Rijndael, AES-The Advanced Encryption Standard, Springer-Verlag Berlin Heidelberg, New York (2002) 31-50.

UML Model and Implementation of the IEEE 802.11 Authentication and Privacy Services using Elliptic Curves Cryptosystems

Rocío A. Aldeco Pérez, Miguel A. León Chávez

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
14 sur y Av. San Claudio, C.U., Puebla, Pue.
Tel: 2295500 ext. 7213 Fax: 2295672
raldeco@mail.cs.buap.mx, mleon@cs.buap.mx

Abstract. In the last years a spectacular development in the mobile communications has been produced, specifically in Wireless Local Area Networks (WLANs). The main function of this kind of networks is provide connectivity and access to the traditional wired networks, such as Ethernet, like it will be an extension of these, but with more mobility and flexibility. One of the most important standards in this area is the IEEE 802.11. This standard provides users with three security services (authentication, privacy, and integrity) based on WEP (Wired Equivalent Privacy) protocol and CRC32. WEP, in turn, is based on the RC4 symmetric algorithm, which is a quite powerful crypto algorithm. Nevertheless, WEP takes a poor approach for using it. This paper proposes the usage of the Elliptic Curves Cryptosystems (ECC) in order to implement the authentication and privacy services into the IEEE 802.11, and presents the modelling of these services using UML (Unified Modelling Language).

1 Introduction

The IEEE 802.11 standard [1] defines two OSI layers, Physical (*PHY*) and Data Link (*DLL*), the latter is divided in two sublayers, Logical Link Control (*LLC*) and Medium Access Control (*MAC*). The *MAC* sublayer provides upper layers with three main services, control of the medium access, mobility, and security. Two access methods are defined by the *MAC* sublayer, one centralised (Point Coordination Function, *PCF*) and another distributed (Distributed Coordination Function, *DCF*). *PCF* uses an access scheme based on polling, where the Access Point (*AP*) acts as Point Coordinator (*PC*). The *PC* cyclically polls all the stations and grants them rights to transmit.

DCF is based on Carrier Sense Multiple Access with Collision Avoidance (*CSMA/CA*) which is mandatory and has to be presented in all stations. The mobility services are provided by the Distribution System, such as association, reassociation, and disassociation.

The MAC sublayer provides users with three security services, authentication, integrity, and privacy. However, the authentication scheme is quite weak [2]. The integrity scheme is restricted to use the CRC-32 algorithm, and the privacy scheme, named Wired Equivalent Privacy (WEP), does not carry out its purpose [3,4,5]. The new version of the standard, i.e. IEEE 802.11i, will be based on 802.x and extensible authentication protocol (EAP). In the long term, 802.11i might provide a framework for using the Advanced Encryption Standard (AES) [2]. The goal of this paper is to use the Elliptic Curves Cryptosystems (ECC) to implement the authentication and privacy services into the PCF method of the IEEE 802.11 standard.

ECC are public-key mechanisms based on the elliptic curve discrete logarithm problem, whose best known algorithms run in exponential time. This means that a desired security level can be attained with smaller keys in ECC than in other public-key mechanisms, such as RSA.

In order to implement the authentication and privacy security services into IEEE 802.11 using ECC, this paper presents the modelling of the authentication and privacy services using UML (Unified Modelling Language) and the implementation of this algorithms in C++.

The rest of the paper is organized as follows, section II discusses the security services of the IEEE 802.11 standard. ECC are presented in section III, the UML model of the authentication and privacy services are presented in section IV, section V presents a brief introduction about the issues of the implementation in C++, conclusions and future research work are presented in section VI.

2 IEEE 802.11 Security Services

The IEEE 802.11 standard provides upper layers with three security services, authentication, privacy, and integrity. For authentication it uses the WEP protocol. WEP encrypts the data frames with the RC4 algorithm, each frame has to be encrypted with a different key, which consists of the field IV (Initialization Vector, 24 bits) and the WEP shared key (40 bits) concatenated forming a 64 bits key. This key is used to create the RC4 key stream, it takes one bit of this stream and one bit of the original message then it applies an XOR forming a cipher bit [1].

In October 2000, it was shown the vulnerabilities of WEP [3] using any size of key. Fluhrer, Mantin, and Shamir [4,5] describe how WEP takes a poor approach for using RC4 algorithm.

The integrity is provided by the CRC-32 mechanism and it is not a cryptographic mechanism, reason why this service is not guaranteed. The authentication service is implemented by the AP, since it only has to accept frames from the stations previously authenticated.

The IEEE 802.11 specifies two kinds of authentication.

Open System authentication: This is a null authentication algorithm. It accepts all the stations.

Shared Key Authentication: The shared key is the same used in WEP. The AP sends a plaintext exchange to the station; the station encrypts it and returns it to the AP. The

AP does the same and compares both results. If they are equal it allows the access, otherwise returns error [1]. Fig. 1 shows the sequence diagram of the authentication and association services. Park and Dicci [3] discuss the authentication problem in these networks, caused by the bad management and distribution keys.

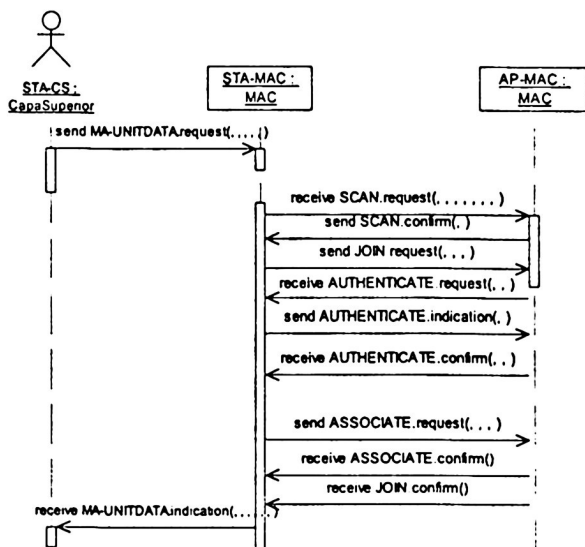


Fig. 1. Authentication and Association Services.

3 Elliptic Curve Cryptosystems

ECC are public-key mechanisms based on the elliptic curve discrete logarithm problem, whose best known algorithms run in exponential time. This means that a desired security level can be attained with smaller keys in ECC than in other public-key mechanisms, such as RSA.

Let q be a prime number, and let F_q denote a finite field (integers modulus n , Galois field, rational numbers, complex numbers) with order q . An elliptic curve E over F_q is defined by the equation:

$$y^2 = x^3 + ax + b \quad (1)$$

where $a, b \in F_q$. The point at infinity denoted by O , is also said to be on the curve. The set of all the points on E is denoted by $E(F_q)$. Elliptic curve points can be added (O plays the role of the neutral element in this operation) but not multiplied. It is, however, possible to perform scalar multiplication, which is another name for repeated addition of the same point. If k is a positive integer and $P \in E(F_q)$, the scalar multiple kP is the result of adding $k - 1$ copies of P to itself.

If $G \in E(F_q)$ then $\langle G \rangle$ denotes the set $\{O, G, 2G, 3G\}$, if we have a point $Q \in \langle G \rangle$, there is a positive integer number k such that $kG=Q$ [6]. The discrete logarithm problem for Elliptic Curves consists of finding the number k starting from G and Q .

Given the huge computational complexity that this problem represents, it is possible to obtain with ECC security levels similar to the provided by other cryptosystems, for the price of operations over finite fields much smaller than the required for the others systems. These operations allows us to use smaller public and private keys that gives us higher velocity, lower memory requirements, and computer power in the implementations of the following algorithms [7].

3.1 ECC Algorithms

3.1.1 System configuration and key generation

First, it chose a finite field $GF(q)$, an elliptic curve E over $GF(q)$ and a point P over E of prime order n , these are the system parameters and are publics [7]. The process for key generation is the follow:

1. Select a random integer number d such that $1 \leq d < n$
2. Compute the point $Q=dP$.
3. The public key is the point Q .
4. The private key is the integer d .

3.1.2 Encryption Scheme:

If M is the plaintext, the next steps are necessities for cipher it

1. Obtain the public key Q .
2. Represent the message M as an element of the field $m \in GF(q)$.
3. Select a random integer number k such that $1 \leq k < n$.
4. Compute the point $(x_1, y_1)=kP$. Compute the point $(x_2, y_2)=kQ$. If $x_2=0$ then go to step 3.
5. Compute $c=m \cdot x_2$.
6. Send the ciphertext (x_1, y_1, c) .

3.1.3 Decryption Scheme:

If we have the ciphertext (x_1, y_1, c) the next steps are necessities for decipher it.

1. Compute the point $(x_2, y_2)=d(x_1, y_1)$, using the private key d .
2. Recover the message m computing $m=c \cdot x_2^{-1}$.

3.1.4 Digital Signature

The next steps are necessities for to sign the message M .

1. With some hash algorithm compute $e=H(M)$.
2. Chose a random integer number k such that $1 \leq k < n$.
3. Compute the point $(x_1, y_1)=kP$ and $r=x_1 \bmod n$.

4. Use the private key for compute $s=k^{-1}(e+rd) \bmod n$.
5. The signature of the message M is (r,s) .

3.1.5 Digital Signature Verification

For verify the digital signature (r,s) in the message M .

1. Obtain the public key Q .
2. If $(r \bmod n)=0$ then reject the signature
3. With the hash algorithm compute $e=H(M)$.
4. Compute $s^{-1} \bmod n$.
5. Compute $u=s^{-1}e \bmod n$ and $v=s^{-1}r \bmod n$.
6. Compute the point $(x_1, y_1)=uP+vQ$.
7. Accept the signature for the message M if and only if $(x_1 \bmod n)=r$.
8. If $r=0$ then the sign equation $s=k^{-1}(e+rd)$ does not involve the private key d , and reject it.

4 UML Model

UML has a Semi-formal semantic meta-model that defines basic modelling concepts (object, class, etc.) and includes well-formed rules expressed as formal constraints. UML has a graphical notation for modelling concepts; it is the standard language for visualizing, specifying, constructing, and documenting the artefacts of a software-intensive system. It can be used with all processes, throughout the development life cycle and across different implementation technologies. UML allows creating several diagrams such as use case, class, sequence, collaboration, etc. These diagrams are the elements of the models. This section presents the analysis and design models of the authentication and privacy services into the PCF mode of the IEEE 802.11 using ECC.

Fig. 2 shows the use case diagram, where the actor is a station, which request any

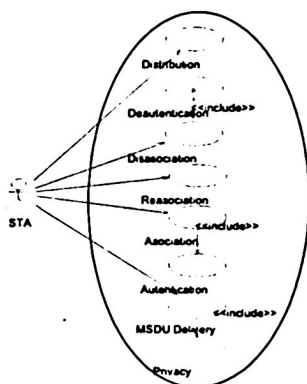


Fig. 2. Use Cases Diagram.

MAC service.

Fig. 3 shows the class diagram that models ECC and figure 4 shows the sequence diagram of encryption and decryption scheme explained in the previous section.

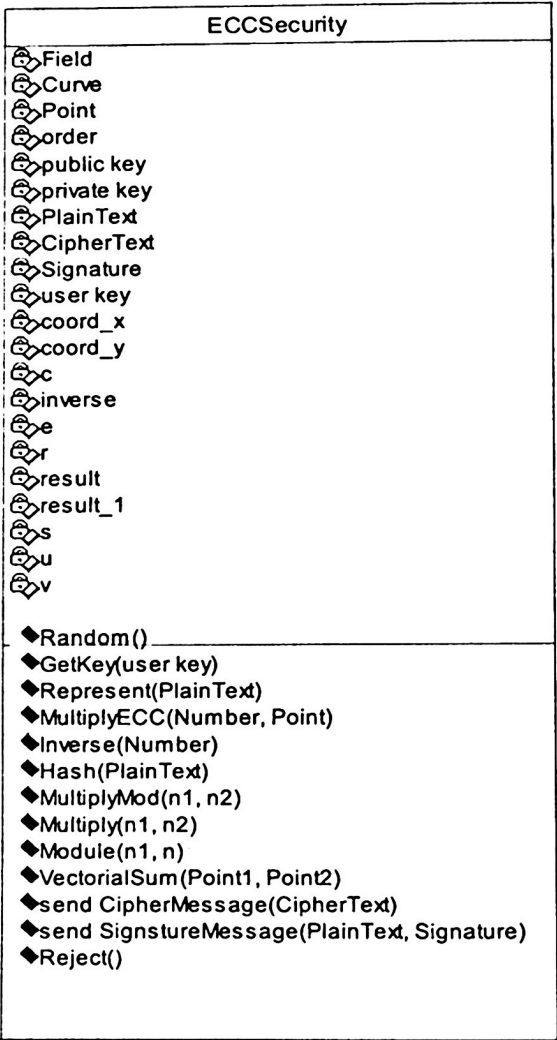


Fig. 3. Class Security ECC.

Both users, A and B, compute their keys (public and private) and then exchange only the public key. After this all the processes continue as before.

Fig. 5 shows the sequence diagram of the signature and verification scheme with ECC. User A signs a message in plaintext with his private key and then sends it to user B.

The user B checks the signature using A's public key. Both users represented in fig. 4 and 5 can be a station or an AP for the encryption – decryption scheme, or they can be a station authenticating to an AP for the signature – verification scheme.

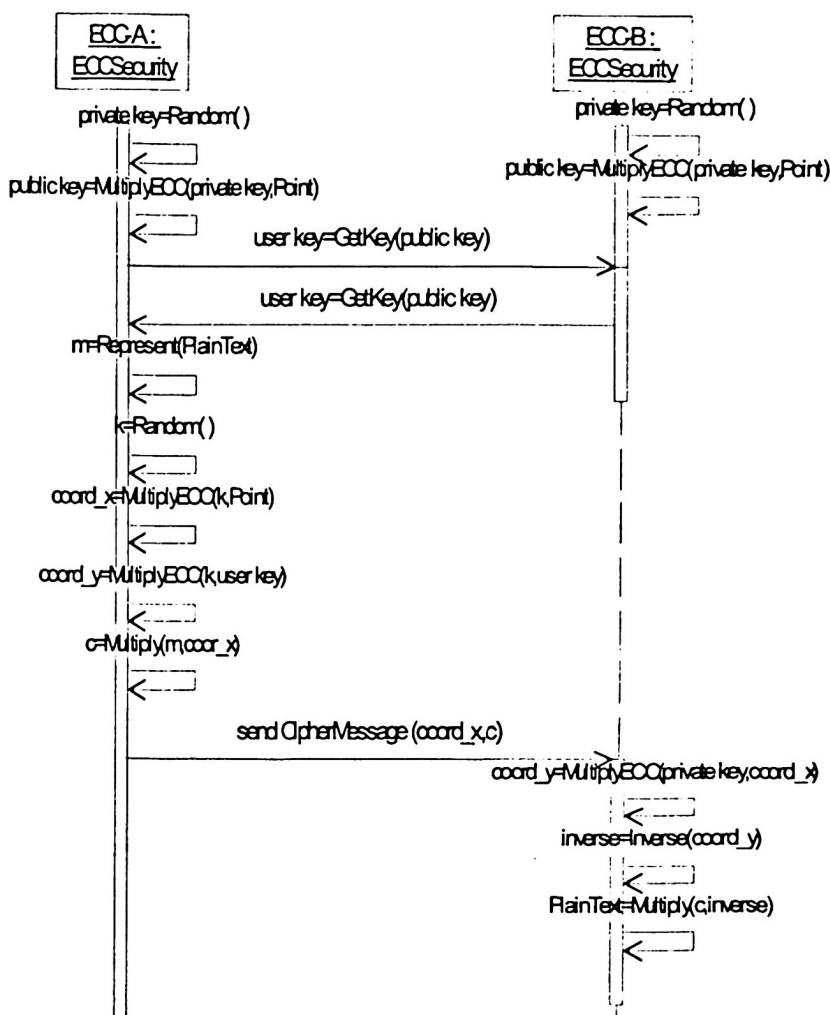


Fig. 4. Encryption – Decryption Scheme.

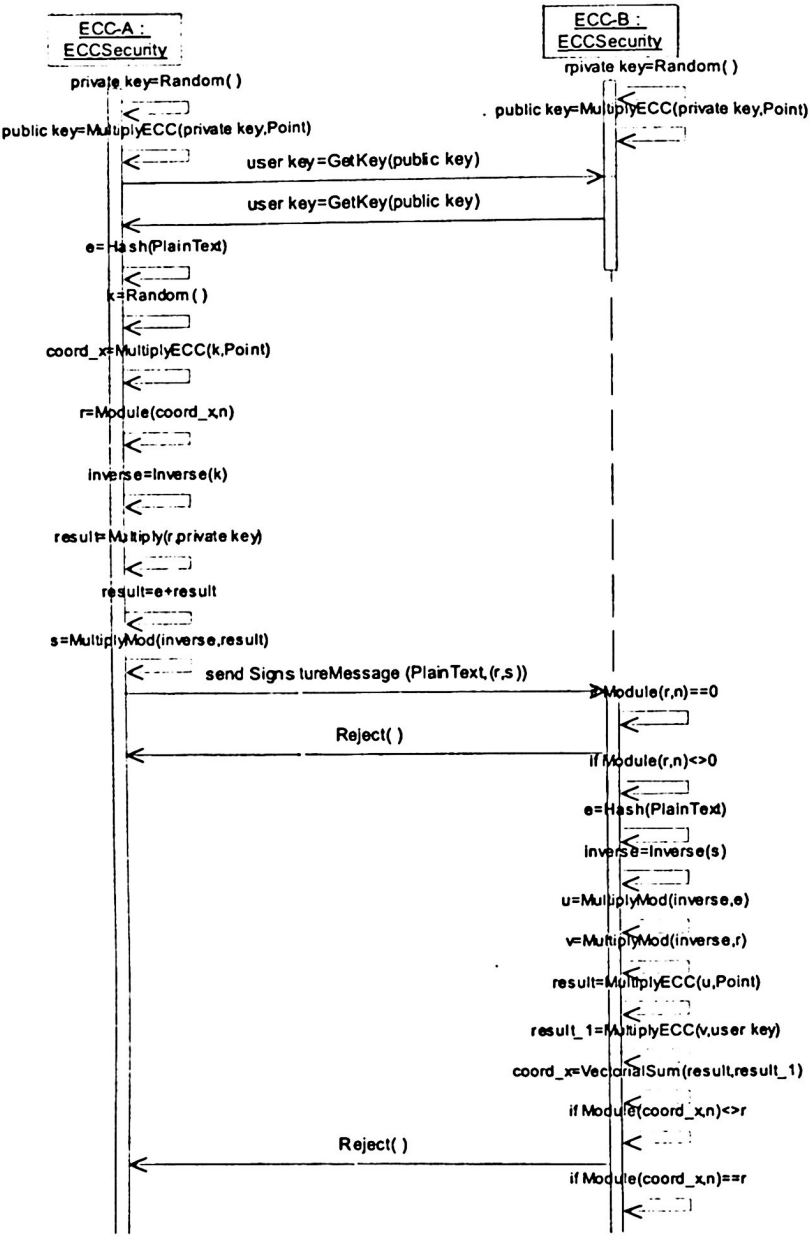


Fig. 5. Signature – Verification Scheme.

5 Implementation

We decided to make the implementation under the programming language C++. Besides to be a great language due to its flexibility, supporting data abstraction and supporting generic programming, also it is an object oriented programming language. The most important issue is, existing a lot of tools, programming in C++, that provide the mathematical background necessary to implement the elliptic curves algorithms.

One of these tools is NTL [8]. NTL is a high-performance, portable C++ library providing data structures and algorithms for manipulating signed, arbitrary length integers, and for vectors, matrices, and polynomials over the integers and over finite fields.

We use NTL for creating our own library for elliptic curves. This library implements all the necessary operations of elliptic curves algorithms (addition, multiplication, inverse, etc.). Consequently, this library doing easier the process of programming.

After that, creates the program that implements both the Encryption – Decryption Scheme and Signature – Verification Scheme.

This program shows all the process modeling in UML. The next step is including these programs in a driver of a wireless card and doing performance test.

6 Conclusions

This paper has discussed the security services defined by the IEEE 802.11 and it has reviewed the vulnerabilities of the authentication, privacy and integrity services. This work has proposed to replace the cryptographic mechanisms implemented in the authentication and privacy services by public key cryptosystems based on Elliptic Curves due to its theoretical efficiency, and that the key size, digital signatures and cipher messages are smallest.

In fact, the cryptosystems based on EC provide the same security level than the systems based on the factorization problem or discrete logarithm problem reducing considerably the number of digits.

ECC can be implemented with efficient in hardware or software and be able to compete in speed with systems like RSA.

The paper has presented the analysis and design models of IEEE 802.11 in PCF mode extended with the authentication and privacy security services using ECC, its implementation in a lower layer, doing performance test and the definition of the key exchange protocol is our future research work.

References

1. IEEE Std 802.11, 1999 Edition. "IEEE ANSI E 802 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1999.

2. Park, J. and Dicoi, D. "WLAN Security: Current and Future". IEEE Internet Computing, September – October, 2003, pp. 60-65.
3. Walker, J. "Unsafe at any key size: An analysis of the WEP encapsulation". IEEE 802.11-00/362. October 2000.
4. Fluhrer, S., Mantin, I., and Shamir, A. "Weaknesses in the Key Scheduling Algorithm of RC4", 8th Annual Workshop on Selected Areas in Cryptography, August 2001.
5. Stubblefield, A., Ioannidis, J., and Rubin A. "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP", Computer Science Dept. Rice University and AT&T Labs – Research, Florham Park, NJ.
6. Jurisic, A. and Menezes, A. "Elliptic Curves and Cryptography: Strong digital signature algorithms" Dr. Dobb's Journal, April 1997.
7. Hankerson, D., Menezes, A., and Vanstone, S. "Guide to Elliptic Curve Cryptography", Springer-Verlag, New York, 2004.
8. Shoups V, NTL: A Library for doing Number Theory (version 5.3.2) available in <http://shoup.net/ntl/>

Construcción de agentes para el TAC SCM Game 2005

Daniel Macías Galindo¹, Fabiola López y López², Darnes Vilariño Ayala²

¹Escuela de Ingeniería en Computación, Universidad Popular Autónoma del Estado de Puebla.

²Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla.
deilan.exe@gmail.com, fabiola.lopez@siu.buap.mx, darnes@cs.buap.mx

Resumen. Con el fin de estudiar mercados dinámicos donde diferentes proveedores y consumidores interactúan, ha sido creada una plataforma que permite la competencia de agentes que trabajan en ambientes de cadenas de producción (The Trading Agent Competition Supply Chain Management Game ó TAC SCM). En ésta competencia, agentes proveedores de computadoras se esmeran en ganar pedidos de clientes y al mismo tiempo en asegurarse que los proveedores les garanticen los componentes básicos que les permitan satisfacer esos pedidos. Al final los agentes ganadores son aquellos que obtengan la máxima ganancia en condiciones cambiantes del mercado. En este artículo se proponen tres agentes con diferentes estrategias capaces de competir en este tipo de juego. Se muestran los resultados de cada agente sobre la plataforma de juego.

1 Introducción

Una cadena de producción se define como el desglose de actividades de flujo y transformación de bienes y productos, desde los componentes básicos hasta la disponibilidad de un producto final en los aparadores [1, 7, 8]. La característica principal de una cadena de producción es que cada actividad puede ser realizada por varias entidades totalmente independientes, situando a una empresa como un eslabón de una cadena. Este esquema implica que mucho del éxito de una empresa para fabricar un bien depende de que tan confiables sean sus proveedores quienes le brindan los componentes básicos para elaborar sus productos. Sólo así, la empresa podrá ofrecer un producto de buena calidad, en el menor tiempo posible, y al costo más bajo. De esta manera se tiene que lograr una administración que genere resultados rápidos, económicos y confiables, que puedan ajustarse a las variaciones en el mercado más complejo, dependiendo de la dimensión del mismo.

Las cadenas de producción observadas en la economía de hoy en día son estáticas, puesto que se basan en relaciones duraderas entre cliente y proveedor. Esta forma de trabajo no se ha cambiado debido a la complejidad de organizar a un amplio sector industrial local, por lo tanto es mucho más complejo organizar un sector económico mundial. La computación promueve implementar tecnologías que permitan prácticas más flexibles y dinámicas, ayudando al cliente a seleccionar al proveedor que le garantice ser la mejor opción, basándose en la información generada por el mercado.

Los agentes de software, por sus características de fácil adaptación al medio ambiente han sido considerados como elementos claves en el modelo de implementación de procesos productivos, que semejen a las cadenas de producción. Un agente es un ente computacional, el cual puede describirse por sus propiedades, que lo diferencian de programas convencionales [2]:

- Es autónomo, porque para actuar no necesita la intervención de un usuario.
- Tiene cierto grado de inteligencia, que va desde reglas hasta motores de aprendizaje que le permiten adaptarse al medio.
- Tiene habilidad social, por lo que puede comunicarse con usuarios, el sistema y otros agentes, según lo requiera.
- Tiene capacidad cooperativa, para realizar tareas complejas en equipo.

Los Laboratorios de Administración de Cadenas de Proveedores de la Universidad de Carnegie Mellon y del Instituto Sueco de Ciencias Computacionales (SICS) han estudiado los problemas presentados en las cadenas de producción desde hace más de seis años [3,4,5]. Para ello se dieron a la tarea de implementar una plataforma que se encarga de simular ambientes de cadenas de producción, donde las condiciones del mercado varían continuamente. Para ello también desarrollaron un conjunto de herramientas que permiten desarrollar agentes que participen en las simulaciones. Es así como nace el TAC (acrónimo de The Trading Agent Competition). Este grupo de investigación realiza cada año un concurso, en el cual invita a universidades y empresas interesadas en el tema de las cadenas de producción, para la implementación de un agente que participe en las simulaciones del TAC SCM (Supply Chain Management).

Este concurso propone la competencia entre seis agentes durante un juego de 55 minutos, donde el agente que obtenga la suma económica más alta es el ganador. Cada agente simula ser el administrador de una fábrica ensambladora de computadoras que debe satisfacer las órdenes de diferentes clientes respecto a equipos de cómputo. Cada agente puede producir 16 tipos de computadora y debe ser capaz de decidir qué órdenes aceptar en base a sus capacidades de producción y al inventario disponible. Los componentes básicos son obtenidos a través de diferentes proveedores los cuales no siempre están disponibles. Cada agente emplea las más diversas estrategias para seleccionar a sus clientes, solicitar componentes a los proveedores y ensamblar los equipos, de tal manera que se tengan mayores ganancias que pérdidas. Por ello, el problema general no necesariamente se puede enmarcar en una de las áreas específicas de la optimización sino que envuelve varias de ellas, que incluyen teoría de juego, programación dinámica y simulación.

El objetivo de este artículo es describir la competencia, los elementos que intervienen en la misma y presentar tres diferentes estrategias que permitan a un agente competir en el TAC SCM 2005. Se presentan los resultados de cada estrategia cuando el agente compitió con los 5 agentes presentados por la plataforma.

El artículo se organiza de la siguiente manera. En la sección 1 se muestra una breve reseña del concurso TAC SCM. La sección 2 profundiza la descripción del juego, así como la información que se envía entre agentes y el servidor. La sección 3, da una pequeña descripción de los paquetes utilizados para implementar un agente que participe en el TAC SCM. La sección 4 presenta la forma en que hemos abordado la solución del problema. La sección 5 contiene una breve descripción de las tres

diferentes estrategias propuestas, las cuales han sido diseñadas e implementadas. Finalmente, en la sección 6, se proporcionan las observaciones y conclusiones con respecto a este trabajo.

2 El TAC SCM

La competencia se lleva a cabo en una plataforma desarrollada en Java que corre en diferentes servidores localizados en Pittsburgh, Pennsylvania y que puede descargarse de Internet para emplearla de manera local. Esta plataforma se encarga de:

- Albergar a los agentes participantes.
- Activar a los agentes al momento de iniciar un juego TAC, y de removerlo una vez que éste termina.
- Proporcionar la información relevante durante el juego.

Entre los datos que la plataforma proporciona, se encuentran los valores de configuración para identificar al agente, y los datos de trabajo para que éste pueda desempeñarse en la simulación, como son la disponibilidad de componentes de los proveedores, los comportamientos de los clientes, la duración de los días TAC (que generalmente es de 15 segundos), etc. Todos estos valores son cargados en el *día de configuración*, o día 0 de la competencia.

De esta manera, los seis agentes que participan en un juego TAC SCM deberán disputarse las solicitudes de los clientes a través de 220 días TAC. Cada día un agente recibe la siguiente información:

- **De los clientes:** Una lista de requerimientos de cotizaciones de computadoras de diferentes tipos (llamada en lo sucesivo cRFQ); así como los pedidos aceptados por los clientes en respuesta a ofertas que les fueron enviadas el día anterior (si éstas fueron hechas, en lo sucesivo se identificarán como cORD). Además, al recibir su mercancía completa, un cliente puede realizar un depósito a la cuenta del agente, por el monto de la compra de los equipos de cómputo (cDEP).
- **De los proveedores:** Respuesta a las solicitudes del día anterior de cotizaciones de los componentes básicos por parte de proveedores (si éstas fueron enviadas, son las sOFF), así como el envío de componentes que fueron ordenados con anterioridad (sDEL).
- **Del banco:** Estado de cuenta.
- **De la fábrica:** Reporte de inventario: la cantidad de componentes básicos, así como de computadoras que ya han sido ensambladas.

Así mismo, en ese día, el agente deberá estar dotado de métodos que le permitan enviar:

- **A los clientes:** Las solicitudes que el agente considere que pueda satisfacer en tiempo y forma (aOFF).
- **A los proveedores:** Las solicitudes de los componentes que necesita para ensamblar los equipos de cómputo pedidos por el cliente (aRFQ), así como las confirmaciones (aORD) para las ofertas hechas por los proveedores. Al recibir el

pedido enviado por el proveedor, hacer un retiro de su cuenta bancaria para pagar a su proveedor (aRET).

- **A la fábrica:** Las órdenes de ensamblado de equipos (aENS), así como las órdenes de entrega de equipos a sus clientes (fDEL). Estos dos últimos factores son llamados *calendario de producción* y *calendario de entrega* respectivamente. El primero se refiere a que una vez que se tengan los componentes se solicita a la fábrica ensamblar los equipos necesarios para satisfacer una orden, y la segunda a que una vez que se tenga la disponibilidad de la mercancía, se asigne la fecha de entrega para el cliente, la cual debe ser menor o igual a la que se le proporcionó en la cotización. Si la orden no ha sido satisfecha en tiempo, el agente cuenta con cinco días en los cuales obtendrá una penalización del cliente por no entregar la mercancía, antes de que la orden sea cancelada.

En la Figura 1 se muestran de manera resumida las actividades diarias que se realizan en el TAC SCM. Los círculos de color gris contienen mensajes que el agente recibe de las diferentes entidades del juego; los círculos en color blanco contienen los mensajes que el agente envía a estas entidades.

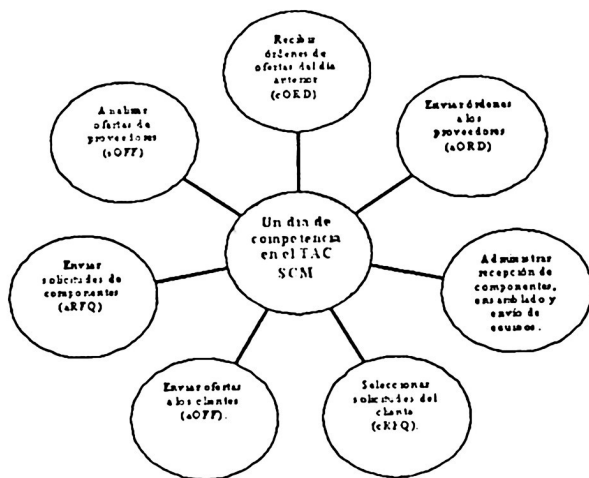


Fig. 1. Actividades que el agente realiza diariamente durante el TAC SCM.

Cada agente tiene su propio almacén de componentes y de equipos, dentro de la fábrica, la cual se dedica al ensamblaje de las computadoras y está limitada por la cantidad de ciclos de trabajo máximos que puede emplear para ensamblar los equipos de cómputo (generalmente 2000 e igualmente dados a conocer en el día 0).

Hay cuatro componentes para tener una PC ensamblada y lista para su venta: CPU, Tarjeta Madre, Memoria y Disco Duro, cada una de ellas son de dos tipos diferentes. Además se disponen de restricciones para el ensamblaje respecto a los CPU y las tarjetas madre, con los cuales se obtienen 16 configuraciones de equipos diferentes.

El juego representa una amplia gama de situaciones de provisión de servicios. El reto del juego está en establecer una estrategia que se ajuste a los cambios en cada simulación, donde el agente demuestre sus habilidades para reaccionar a las variaciones que presentan tanto el posible cliente, como el proveedor, así como adaptarse a las estrategias adoptadas por los otros agentes en competencia.

En la Figura 2, se describe más detalladamente el paso de mensajes de manera ideal en el juego TAC SCM. Los cuadros en negro representan las entidades externas al agente, los cuadros grises al agente y a sus componentes; las flechas indican el paso de mensajes de la entidad emisora a la receptora; los colores de las flechas van de claro a oscuro, entre más claro, es un mensaje que ocurre antes que un mensaje de color oscuro.

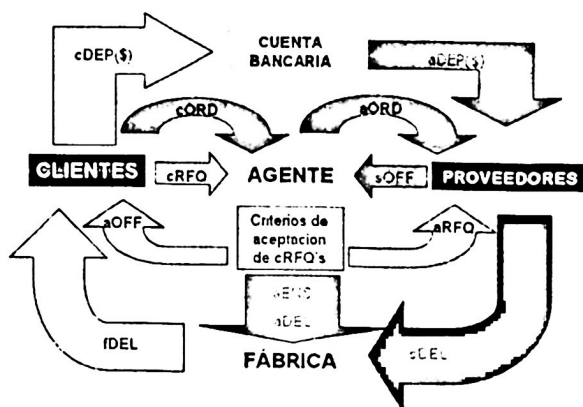


Fig. 2. Diagrama de las actividades que se realizan en el TAC SCM.

Los agentes en este medio ambiente, entonces, deben ser capaces de realizar las siguientes tareas en cada día TAC:

1. Negociar contratos de proveedores.
2. Buscar satisfacer las órdenes del cliente, ofreciéndole los mejores precios en el menor tiempo de entrega.
3. Administrar actividades diarias de ensamblaje y envío dentro de la fábrica.

3 Paquetes para el desarrollo del agente

La paquetería para implementar un agente capaz de competir en el TAC SCM se encuentra disponible al público en el sitio Web del TAC [6]. A continuación se realiza una breve descripción de lo que se puede encontrar en cada paquete.

3.1. Paquete se.sics.tasim.props

Este paquete es el encargado de recepción-envío de mensajes entre el agente y el servidor de juego, es decir entre el agente y las entidades del juego (clientes, proveedores y la fábrica). Cada mensaje está formado por el nombre del emisor, del receptor y el contenido. El contenido del mensaje describe el propósito de la comunicación. Cabe destacar que toda la comunicación (exceptuando los nombres del emisor y receptor, cuyo formato es de String) se realiza con números enteros, lo cual permite una comunicación cliente-servidor más fluida y directa.

3.2. Paquete se.sics.tasim.aw

El paquete incluye clases básicas para implementar al agente participante en estas simulaciones.

3.3. Paquete se.sics.tasim.tac03.aw

Incluye, principalmente, el visualizador del usuario diseñado desde el TAC SCM 2003, el cual es mostrado en la figura 3.

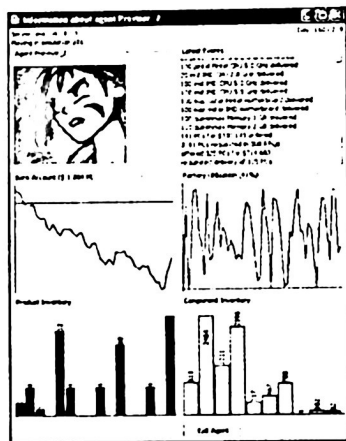


Fig. 3. Visualizador para el usuario del agente Previsor_2.

4 Consideraciones iniciales

La implementación de un agente que se desenvuelva de manera óptima a través de las simulaciones, requiere comenzar a analizar la solución de un problema que puede abordarse usando diferentes estrategias de optimización. Uno de los objetivos de este

proyecto consiste en proveer al agente de una política que le permita obtener el máximo de utilidades en cada juego TAC. En este trabajo se discuten los resultados obtenidos con 3 estrategias que implementan políticas de compra y ventas diferentes, esto nos permitirá confeccionar un agente para la competencia, que pueda de alguna manera cambiar su comportamiento de acuerdo a las condiciones del mercado y de sus ganancias obtenidas.

Los resultados de esta competencia permitirán establecer nuevas relaciones productivas a nivel mundial en la relación entre clientes y proveedores. A continuación se presenta la estructura que conformará a los agentes.

4.1 Diseño del agente

El agente debe enfocarse a tres actividades principalmente en cada uno de los días de la simulación:

- Seleccionar clientes seguros para obtener órdenes de compra.
- Seleccionar sus proveedores para surtir su almacén.
- Administrar las actividades del almacén.

4.1.1. Selección de clientes.

Cada día se recibirá un conjunto de solicitudes de cotizaciones por los clientes (cRFQ). El agente debe ser capaz de seleccionar una población menor de esa lista, considerando que son órdenes que él podrá satisfacer, para lo cual verifica las siguientes condiciones:

- Que los componentes para satisfacer una solicitud estén disponibles en el almacén (dentro de la clase `InventoryStatus` existen métodos para verificar la existencia de componentes y de PCs ensambladas).
- Que los componentes puedan adquirirse en los días sucesivos, preferentemente antes de recibir penalizaciones por entregas retrasadas, si la orden es aceptada (se debe hacer una solicitud, aRFQ, al proveedor correspondiente, apoyado en un método de la clase `SCMAgent`).
- Según se establezca el modelo de solución, se pueden definir uno o más criterios para seleccionar a los clientes; ya sea por fecha de entrega, por precios, cantidad de equipos, o por la combinación de algunos de ellos.

Basándose en esto, el agente debe generar una oferta para cada cliente cuyo pedido puede satisfacer, y al menor precio posible para ganar la orden respecto a los otros agentes, consultado un objeto de la clase `MarketReport` (generada cada 20 días, donde se incluyen el menor y mayor precio ofertado para un tipo de PC). Hay que tomar en cuenta que un cliente aceptará una oferta si ésta viene por la cantidad requerida, si se entrega antes o en la fecha requerida, y si el precio ofrecido es el menor de todas las ofertas hechas por los agentes.

4.1.2. Negociación con proveedores.

Para ello se realizará una planeación de compras, en la cual se deben tomar en cuenta los siguientes factores:

- Los pedidos de los clientes que están pendientes. (Dependiendo del momento en que se haga la compra, sería un objeto de la clase **RFQBundle** o de **OrderBundle**).
- Los pedidos de componentes básicos a los proveedores que aún no han sido satisfechos,
- El gasto que el agente realizará para adquirir estos componentes, (se puede analizar un objeto de la clase **BankStatus**).

Hay que aclarar que esta planeación tiene que contemplar el panorama futuro, y predecir los comportamientos del mercado. Entonces la planeación de compras consiste en que el agente:

- Decide qué comprar (según órdenes pendientes, almacén y estado de su cuenta).
- Decide a quién comprar (según la factibilidad de que el proveedor pueda satisfacer la orden, que se adquiere analizando la preferencia que tiene cada proveedor respecto hacia el agente).
- Decide en qué momento comprar (si lo hace antes de recibir las confirmaciones del cliente, o en el momento de recibirlas).

4.1.3. Administración del almacén.

El agente debe estar capacitado para manejar sus propias actividades de ensamblaje, las cuales:

- Dependen de los componentes en existencia en almacén (hay un método en la clase **SCMAgent**, llamado *addProductionSchedule*, donde se hace este análisis y si pueden armarse la cantidad especificada de PCs, ésta orden se añade al ensamblado del día siguiente en la fábrica).
- Deben satisfacer las órdenes de compra ya hechas de los clientes al agente (igualmente, con el método *addDeliverySchedule*, se analiza la posibilidad de que un pedido completo sea enviado al cliente al día siguiente de que se da la orden).

4.1.4. Administración de la cuenta bancaria.

Con el banco solamente se tendrá la interacción para conocer el estado de cuenta, y saber si el agente es capaz de comprar más componentes, y al final, si es el ganador de la ronda.

5 Estrategias implementadas

Hasta el momento se han implementado tres agentes, con dos estrategias completamente diferentes, con los cuales se ha intentado conocer el funcionamiento de la plataforma, aprovechar la información que se envía en los mensajes y agregar

más situaciones que no han sido contempladas. Todo esto con la intención de tener un agente fuerte, versátil e inteligente para la competencia mundial.

A continuación se discutirán brevemente estas estrategias, y se hará una comparativa de los resultados obtenidos con cada agente.

5.1 Agente Loco_Avorazado

Este agente es la primera implementación realizada para comenzar a entender el funcionamiento de la plataforma. Su funcionamiento se puede resumir en los siguientes puntos:

- Selecciona a los clientes tomando el día de entrega de la orden; éste debe ser igual a ocho días después del día TAC actual.
- Oferta las solicitudes de los clientes al 95% del precio sugerido por el mismo cliente.
- Solicita los componentes el mismo día que envía las ofertas a los clientes.
- Selecciona a su proveedor de manera aleatoria.

RESULTADOS:

- Toma el 10% de las solicitudes totales enviadas cada día.
- Se utiliza en promedio el 24% de la fábrica.
- El cliente prefiere a este agente en el 96% de las ofertas hechas.
- Ganancias: de \$4 M a \$8.2 M.

Result for game 197@localhost played at 2005-03-25 01:23:00

Played with TAC SCM 04 rules (February 2004 specification)

Player	Revenue	Interest	Costs				Margin 1	Margin 2	Result
			Material	Storage	Penalty				
Dummy	82 785 964	1 058 250	57 931 535	231 890	373 619	1%	30%	31%	25 307 170
Dummy-2	84 562 422	1 065 005	59 447 116	239 713	762 266	1%	29%	30%	25 178 332
Dummy-4	86 274 490	1 064 240	61 070 658	247 744	963 993	2%	29%	29%	25 056 335
Dummy-3	80 504 809	1 034 900	56 238 763	228 033	793 775	1%	30%	30%	24 279 138
Dummy-5	82 253 921	928 198	58 402 467	247 182	3 006 001	5%	29%	26%	21 526 469
Loco_Avorazado	42 706 142	319 508	30 791 591	96 723	4 359 310	12%	28%	18%	7 778 026

Download game data [here](#)

Player	Orders	Utilization	Deliveries (on time/late/missed)	DPerf
Dummy	3846	49%	3715 / 131 / 0	97%
Dummy-2	3910	50%	3629 / 281 / 0	93%
Dummy-4	4029	51%	3738 / 291 / 0	93%
Dummy-3	3797	48%	3533 / 264 / 0	93%
Dummy-5	3789	49%	3107 / 682 / 0	82%
Loco_Avorazado	2210	27%	1711 / 354 / 145	77%

Fig. 4. Resultados del agente Loco_Avorazado.

5.2 Agente Previsor_2

- Selecciona clientes respecto a una función entre precio sugerido y cantidad de equipos a comprar.
- Su inventario se obtiene *a priori*, entonces tiene que asignar de manera óptima tanto éste como su trabajo en la fábrica: por lo que se crea un problema de optimización: maximizar ganancias, sujetándose a la cantidad de componentes en inventario y ciclos de trabajo libres en la fábrica.
- Funciona con un algoritmo selectivo, que ordena los pedidos y decide qué órdenes pueden ser satisfechas con nuestras limitantes (inventario y fábrica).
- La producción de los equipos de cómputo se realiza inmediatamente que se envían las ofertas al cliente.

RESULTADOS:

- Aceptación del cliente del 99%, debido al precio ofertado del 92%.
- Existe un remanente en inventario tanto de componentes como de equipos no ensamblados.
- Los equipos que no se entregan en tiempo, nunca son enviados a sus clientes (posiblemente debido a un error en el esquema de análisis de entregas).
- Ganancias variables, desde saldo negativo hasta \$14 M.
- Su falta de constancia no es suficiente para competir contra los *Dummies* (agentes demo que corren en la plataforma).

Result for game 136@localhost played at 2005-03-02 05:47:00

Played with TAC SCM 04 rules (February 2004 specification).

Player	Revenue	Interest	Costs				Margin 1	Margin 2	Result
			Material	Storage	Penalty				
Dummy-2	74 073 709	657 047	55 017 982	198 405	1 801 001	3%	25%	24%	17 713 368
Dummy-3	71 691 196	657 370	52 661 800	200 297	2 556 928	5%	26%	24%	16 929 541
Dummy	72 753 713	640 361	53 898 947	206 941	2 359 649	4%	26%	23%	16 928 537
Dummy-4	68 487 339	603 112	49 963 677	190 860	2 102 179	4%	27%	25%	16 833 735
Dummy-5	71 563 973	589 644	52 468 135	205 140	3 463 654	6%	26%	22%	16 016 688
Previsor_2	90 263 967	104 893	72 671 940	904 350	12 700 665	15% 18%		5%	4 091 905

Download game data [here](#)

Player	Orders	Utilization	Deliveries (on time/late/missed)	DPerf
Dummy-2	3718	44%	3194 / 520 / 4	86%
Dummy-3	3534	43%	2868 / 664 / 2	81%
Dummy	3701	43%	3127 / 533 / 41	84%
Dummy-4	3489	41%	3050 / 417 / 22	87%
Dummy-5	3617	43%	2918 / 682 / 17	81%
Previsor_2	2834	54%	2246 / 0 / 588	79%

Fig. 5. Resultados del agente Previsor_2.

5.3 Agente Previsor_3

- Su operación es similar a Previsor_2.
- Implementa la compra de los componentes de acuerdo al inventario y a cuotas máximas y mínimas, así como la siguiente regla: tomando en cuenta 4 diferentes procesadores, se compran menos procesadores que los demás componentes.
- Con esto ajusta sus compras, quedando con un inventario todavía menor al final del juego que Previsor_2.
- Limita compra de componentes y ofertas en el momento adecuado antes del final del juego.

RESULTADOS:

- Ganancias de \$29 a \$39 M, primer lugar.
- Todos los pedidos se entregan en tiempo y forma, no hay penalizaciones.
- Remanente de inventario en componentes, más no en equipos ensamblados.
- Precios ofertados al cliente, fijos.

Result for game 181@localhost played at 2005-03-11 03:02:00

Played with TAC SCM 04 rules (February 2004 specification)

Player	Revenue	Interest	Costs			Margin 1	Margin 2	Result
			Material	Storage	Penalty			
Previsor 3	127 286 487	1 120 193	91 002 270	183 919	0	0% 28%	29%	37 220 491
Dummy-3	70 341 194	807 564	48 507 512	140 282	1 607 982	3% 31%	30%	20 892 982
Dummy	70 174 845	816 435	48 173 347	139 362	1 864 515	4% 31%	30%	20 814 056
Dummy-5	70 010 335	805 223	48 171 366	138 945	1 728 061	3% 31%	30%	20 777 186
Dummy-4	71 904 724	775 918	49 765 980	143 578	2 119 062	4% 31%	29%	20 652 022
Dummy-2	67 717 514	757 742	46 409 527	133 241	1 412 388	3% 31%	30%	20 520 100

Download game data [here](#)

Player	Orders	Utilization	Deliveries (on time/late/missed)	DPerf
Previsor_3	3530	78%	3530 / 0 / 0	100%
Dummy-3	3626	42%	3174 / 452 / 0	88%
Dummy	3623	42%	3056 / 567 / 0	84%
Dummy-5	3617	41%	3107 / 510 / 0	86%
Dummy-4	3750	43%	3205 / 526 / 19	85%
Dummy-2	3523	40%	3094 / 419 / 10	88%

Fig. 6. Resultados del agente Previsor_3.

6 Conclusiones

Para analizar el comportamiento de las estrategias propuestas, se lanzaron los 3 agentes de manera local a competir con los *dummies* (agentes de prueba) que posee la

plataforma. Los resultados de cada uno de los juegos se muestran en las figuras 4, 5 y 6. Se comprobó que la estrategia de compra y venta que se implementó en Loco_Avorazado no permite ganancias sustanciales en el mercado cambiante, por su parte para encontrar una solución del modelo propuesto en Previsor, se utilizó un algoritmo de Monte Carlo, sin embargo no siempre se encontró la solución en el tiempo adecuado (antes de 15 segundos), por lo que se perdieron un buen número de clientes y se decidió por la estrategia selectiva implementada en Previsor_2.

En Previsor_3 se trató de eliminar los problemas presentados con los otros agentes (Loco_Avorazado y Previsor_2), esto permitió lanzar al agente al primer lugar en la competencia contra los *Dummies*. Sin embargo aún se tienen problemas en el manejo del inventario respecto a computadoras ensambladas y a la cantidad de componentes de determinados tipos, que aún quedan en el almacén, esto permitió el diseño de una nueva estrategia Previsor_Avorazado que se encuentra en fase de implementación.

Próximamente la plataforma en Internet lanzará los agentes ganadores del año anterior, con lo que pretendemos probar todas nuestras estrategias y finalmente proveer al agente de diferentes mecanismos que le permitan modificar su comportamiento de acuerdo al mercado cambiante y al número de ordenes que logre ganar.

Los resultados han sido satisfactorios, de agentes perdedores logramos diseñar un agente que siempre ocupe el primer lugar. Hasta el momento no se han incluido técnicas fuertes de optimización, Previsor_Avorazado ya considera técnicas de juego con lo que ya estaremos en condiciones de enfrentarnos a los agentes ganadores del año anterior. Por primera vez México participará en dicha competencia mundial.

Referencias

1. Hellriegel, Jackson y Slocum. "Administración: Un enfoque basado en competencias", Editorial Thomson, 9a Edición, pp. 58-59
2. H. Nwana, Software Agents: An Overview, The Knowledge Engineering Review, Volume 11, Number 3, pp. 205-244. 1996.
3. N. M. Sadeh, R. Arunachalam, J. Eriksson, N. Finne and S. Janson, "TAC'03: A Supply Chain Trading Competition", AI Magazine, Spring 2003.
4. Arunachalam, R., Sadeh, N., Eriksson, J., Finne, N., and Janson, S., "Design of the Supply Chain Trading Competition", IJCAI-03 Workshop on "Trading Agent Design and Analysis", Mexico, August 2003.
5. Arunachalam, R., Sadeh, N., Eriksson, J., Finne, N., and Janson, S., "The TAC Supply Chain Management Game", 2003 CMU-CS-03-184 tech. Report
6. Sitio web de la Trading Agent Competition: <http://www.sics.se/tac/page.php?id=1>
7. Gómez Escobar, Ignacio; "La logística en el mercadeo", ABC Formación: http://www.abcformacion.com/contenidos/logistica_0006.htm
8. López, Carlos; "Administración de la cadena de abastecimiento": <http://www.gestiopolis.com/canales/gerencial/articulos/26/aca.htm>

Managed Web Services using WS-Manageability

Giner Alor Hernández, José Oscar Olmedo Aguirre

Research and Advanced Studies Center of IPN (CINVESTAV).

Electrical Engineering Department. Computing Section.

Av. Instituto Politécnico Nacional 2508, Col San Pedro Zacatenco. 07360 México, D. F.

e-mail: ginalor@computacion.cs.cinvestav.mx, oolmedo@delta.cs.cinvestav.mx

Abstract. As Web services become pervasive and critical to business operations, the task of managing Web services and implementations of the Web services architecture is imperative to the success of business operations. Web services manageability is defined as a set of capabilities for discovering the existence, availability, health, performance, and usage, as well as the control and configuration of a Web service within the Web services architecture. In this work, we have developed an initial prototype of the WS-Manageability specification, which illustrates how manageability functionality can be seamlessly added to a collection of Web services in a totally non-invasive manner to the services being managed. A basic manager is also included, which allows pluggable manageability aspects of the managed web services to be configured.

1 Introduction

The emergence of Service-Oriented Architectures (SOA) [1] and industry commitment to Web services and Grid computing [2] requires new management capabilities in order to truly achieve the visions and goals of these initiatives. Because, both Web services and Grid environments, leverage the loosely coupled nature of services, deployed across heterogeneous platforms and ownership domains and can dynamically discover and interact with each other during solution design or execution, the manageability of such environments is more critical and challenging to the business they support. Web services can also cross administrative domains leading to multiple aspects of control requiring a greater emphasis on declarative and real-time management through the use of management policies and service level agreements. Having this into account, we have developed an initial prototype of the WS-Manageability specification, which is able to manage resources (as Web services) remotely from a central location to make management practical. In our prototype the underlying technology used to manage the resources is JMX (Java Management eXtension) [3], but conceptually could be extended to support other management technologies such as CIM (Common Information Model) [4] and SNMP (Simple Network Management Protocol) [5].

The rest of this paper is structured as follows. In the next section we provide the main concepts of WS-Manageability specification. In the following sections we present the architecture of our prototype developed and describe the functionality and its

main components. Then we describe the future directions and the work to do. Finally, we emphasize the contributions of our work.

2 Web Services Manageability

The WS-Manageability specification states that Web services management concerns are to be partitioned into "Topics". A topic covers a functional capability that supports management of a particular problem or management domain. For example, the state management for a manageable resource is a management topic. Defining manageability within topics enables incremental and modular development and support of manageability capabilities. The functional capability of a topic can be described, or modeled using a combination of three aspects: properties, operations, and events. Next, we describe the five topics from WS-Manageability specification, which are:

- **Identification:** provides the functional capability to uniquely identify the resource being managed. This static information may include properties that are not required for unique identification, including descriptive and semantic information.
- **State:** provides the functional capability to manage the actual operational state of a resource (i.e. up and down states in a lifecycle) [6]. The State topic defines properties about the operational state, operations to influence a change in state and events indicating when a state change has occurred.
- **Configuration:** provides the functional capability to manage the collection of properties whose values may influence the behavior of a resource. These properties may be changed by the resource or may be changed by the manager which may cause behavior changes in the resource [6]. Operations to access and change the configuration along with events indicating configuration changes are also part of the configuration topic.
- **Metrics:** provides the functional capability to manage metrics for a resource. The metric functional capability includes reset operations for metrics, and metric collection controls. Metrics are raw atomic, unambiguous, quantifiable information. The value of the metric captures the information at a point in time. Generally, these values are numeric, but may be strings as well. Metrics can be contrasted with derived metrics that are calculated using a formula and metrics [6]. For example, average response time during the last hour of execution is an example of a derived metric.
- **Relationships:** provides the capability to query associations that the resource (e.g., endpoint) participates in with other resources. Relationships cover the associations that may exist between a resource and other resources of the same or disparate type. Relationships are important to management for problem isolation, root cause analysis, and impact analysis where resources are related in some way. Relationship types that are defined between resources can have common or very specific semantics, requirements, and implications for management [6]. For this reason it will be common to have topics that include operations that will create relationships that are viewed through this topic.

Within each of these topics are the details which make up the specification. These details are divided for each topic into the following three "aspects".

- **Properties:** provide a way to advertise or surface state information about a resource (e.g., endpoint). They are expressed as named elements in an XML infoset and may be either of a simple type or a complex type. Properties are manipulated directly through operations defined as part of the management interface (e.g. get/set operations) or by events that occur outside of the influence of the manager (e.g. a configuration file being updated). Property change events may be emitted when a property's value is changed. Which property changes cause events to be emitted must be described. Emission of property change events may also be controllable by a manager to permit pausing and resuming the emission of property change events.
- **Operations:** are methods to control the resource or to retrieve state information. Operations may cause temporary or permanent changes in the Web service's behavior.
- **Events:** describe the indications of changes with respect to the resource. An event description is a set of information that describes the change. The information structure of an event description is defined by a named complex type. Notifications are messages containing the event descriptions that are transported to an interested party.

In the next section, we describe the architecture and its components of a prototype that we developed, which covers the main concepts from WS-Manageability specification such as identification, state, configuration, metrics, properties and operations.

3 Web Services Manageability Prototype

We have developed a prototype, which clearly illustrates the spirit and intent of the WS-Manageability specification. The underlying technology used to the implementation is JMX (Java Management eXtension). The JMX architecture consists of three levels: instrumentation, agent, and distributed services. JMX provides interfaces and services adequate to monitoring and management systems requirements [7]. This functionality involves abstracting resources by using components called MBeans (Managed Beans) and remote instrumented resources accessibility through JMX connectors. An MBean is a Java object that represents a manageable resource, such as an application, a service, a component, or a device [7].

The main component of our architecture is a JMX Bridge, which acts as a bridge between the world of resources managed by JMX and web services. In our prototype, the Web services interfaces to JMX are available. Rather than provide a JMX specific web service interface, our prototype provides a web service interface to a manageable resource. Under this approach, the resources can be implemented on different technologies because only is necessary to define a web service interface for a resource. For doing this, we use MBeans to represent a resource being managed. The JMX

Bridge is given information which identifies or describes the MBean instance which represents a specific managed resource. The JMX bridge generates a WSDL document describing the web service, a Java class which acts as the web service implementation, and an Axis deployment descriptor. The resulting Java class must then be compiled, made available to an application server and deployed to Axis as a web service. The generated WSDL can be used to make dynamic calls, validate static calls, or generate a client proxy class. Fig. 1 shows the general architecture of our prototype. In Fig. 1, a generic management handler has been implemented and added to application service's handler chains to gathers statistics.

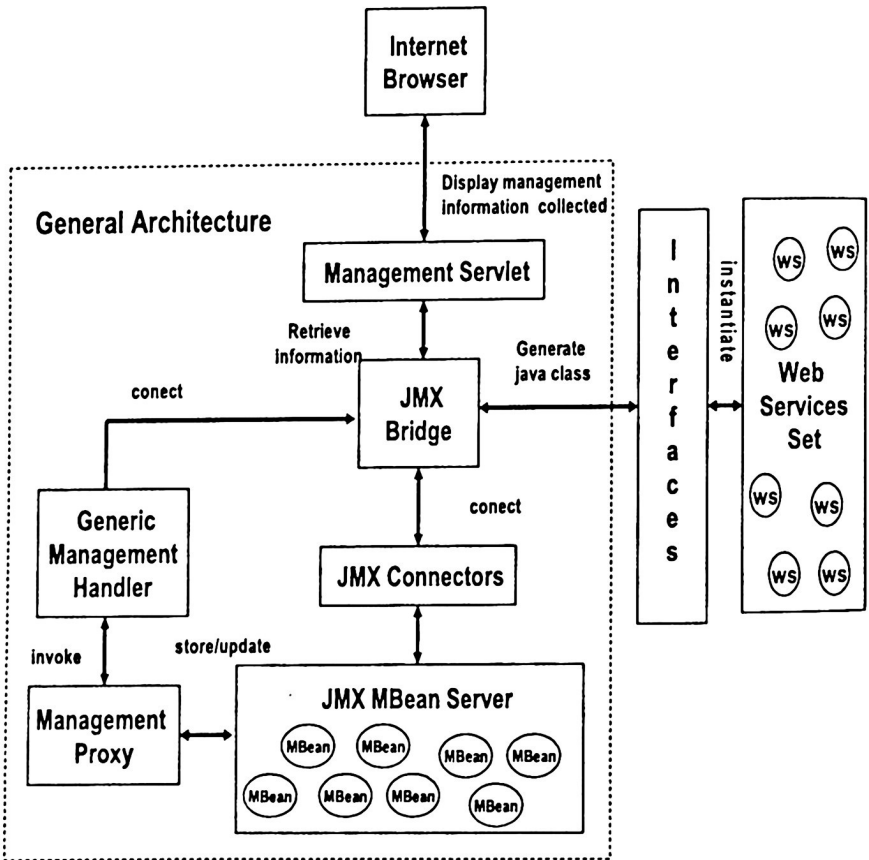


Fig. 1 General Architecture of prototype based on WS-Manageability.

The handler invokes a management proxy which stores, or updates the statistics in MBeans which are instantiated and managed by the JMX compliant MBean server.

Fig. 2 shows a screenshot where the JMX compliant MBean server deploys the stored resources as web services. Also, a management servlet was developed which provides a view of management information collected. The MBean Server and its supporting classes are instantiated by the application server's class loader, which allows this single MBean Server instance to gather MBeans for all management instrumented web services.

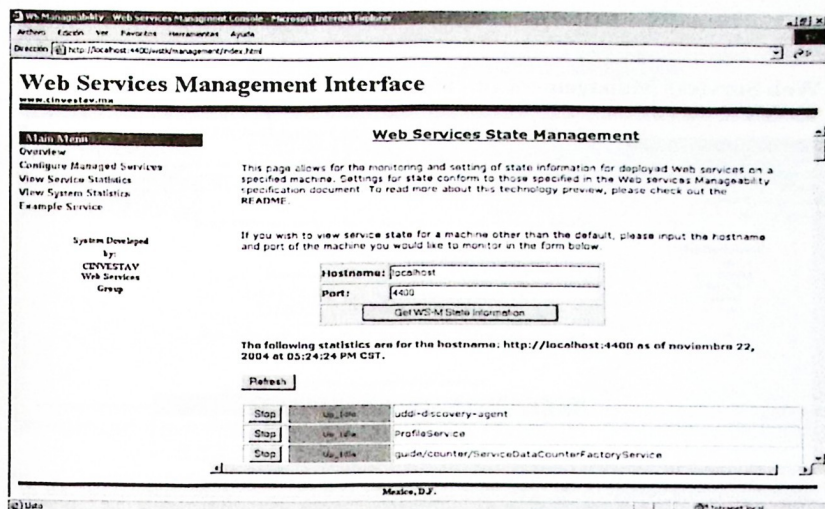


Fig. 2 The JMX compliant MBean Server deploying registered web services.

The most significant of this work is a JMX-based management web service which allows to store MBeans within a JMX MBean Server which can be accessed and manipulated from an AXIS-enabled remote console. Under this approach it is possible to monitor the status of JMX-enabled services and components through an AXIS (SOAP) client interface. The MBean server tracks global information and statistics about web services. For doing this, a web-based interface was developed, which displays this data based on a hostname and port pair specified in a simple form embedded in a servlet. Collectively, this functionality represents the direction towards active management of JMX-enabled data and application resources over a standard AXIS (SOAP) interface. Furthermore, MBean tracks global information regarding the configuration and use of the server for which information is being requested. This global information can be tracked and retrieved by a remote console for the purpose of monitoring and remotely managing application servers and web services in an enterprise environment. Fig. 3 shows a screenshot of the kind of information that JMX compliant MBean Server deploys. Among the kind of information collected in the JMX compliant MBean Server is included the following:

- The brand and name of the application server being used
- The version of the server being used
- The hostname of the server
- The HTTP port number currently in use by the server
- Total number of managed SOAP services deployed
- Total number of RPC calls to all managed services combined

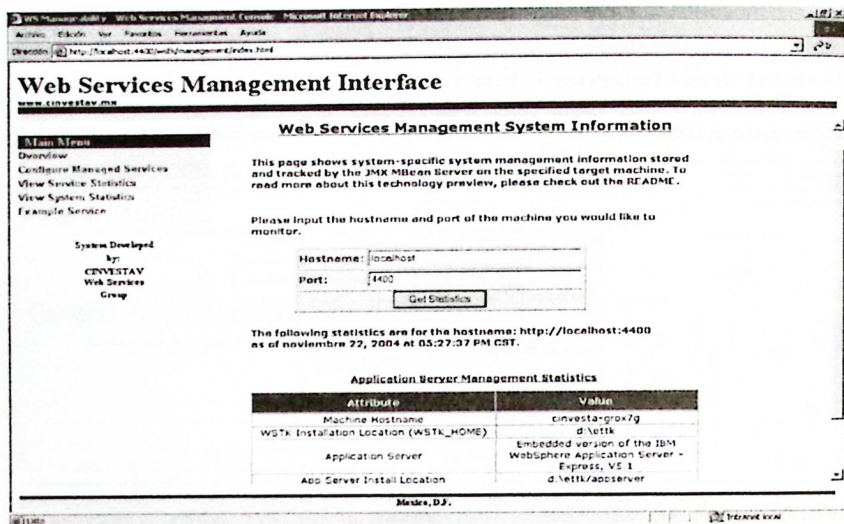


Fig. 3 The JMX compliant MBean Server deploying global information.

This illustrates an approach to managing web services by instrumenting AXIS through the use of handlers to provide a JMX-based systems management interface. In this case, AXIS handlers were developed and added to the handler chains of our various web services to allow statistics about those services to be gathered. The JMX MBean Server which tracks the service statistics is instantiated globally within the application server's JVM allowing statistics to be tracked across all instrumented web applications. Fig. 4 shows a screenshot of the kind of information that JMX compliant MBean Server deploys when the web services are invoked. Under this approach no code changes to the web services themselves are required to enable this functionality. The handlers simply need to be added to each web service's handler chain. Finally, for each instrumented web service invoked, the following four statistics are tracked:

- Current state
- Total number of successful invocations
- Number of failed invocations (request received the server but resulted in an exception)

- Average response/transaction time for successful requests

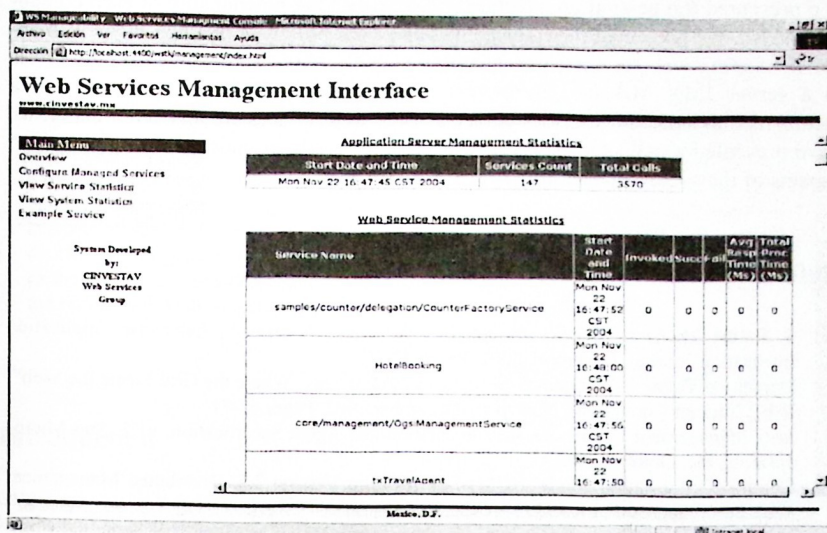


Fig. 4 Graphic interface where the MBean server deploys the current state, total number of successful and failed invocations, and the average response/transaction time for each requests of the stored web services.

4 Future Directions

Our prototype shows only the basics of what a JMX-based management system is capable based on WS-Manageability of. However, many items are possible for future development, including:

- Further implementation of the WS-Manageability specification
- Migrating from AXIS Handlers to Web Services Bus Filters.
- Provide facilities and instrumentation inside web services to expose and illustrate their own management capabilities.
- Extend management capabilities in web services to include automatic and manual operations (the ability to change the behavior, or state of a service), and events and alerts (the ability to send notifications when some criteria is met allowing for more proactive management of the service).
- Provide a mechanism to control which services are managed and how much information should be collected both as a install time and runtime configuration setting.

5 Conclusions

In this work we have presented a prototype of the WS-Manageability specification. We presented the general architecture and described the functionality of each component. Also, we presented the implementation of our prototype in a non-invasive manner to existing web services, which is easy to apply across all deployed Web services in a server JMX MBeans. Furthermore, we propose a JMX MBeans server as a mechanism to store the statistics gathered when web services are invoked. Finally, we have presented a manager interface, which can be easily extended as more topics and aspects of the WS-Manageability specification.

References

1. S. Pallos Michael. "Service Oriented Architecture: A primer". Enterprise Application Integration Journal. December 2001. Pages 32-35.
2. Domenico Talia. "The Open Grid Services Architecture: Where the Grid Meets the Web". IEEE Internet Computing. November-December 2002. Pages 67-71.
3. Java Management Extensions Instrumentation and Agent Specification, v1.2. Sun Microsystems, Inc. October 2002.
4. Common Information Model (CIM) Specification Version 2.2. Distributed Management Task Force, Inc. June 14, 1999
5. M. Feit Sidnie. *SNMP: A Guide to Network Management*. McGraw Hill Series on Computer Communications. September 1, 1994. ISBN: 0070203598.
6. Potts Mark, Sedukhin Igor, Kreger Heather. "The Web Services Manageability Specification 1.0". IBM, Talking Blocks, Computer Associates. September 10, 2003
7. Sun Microsystems, Java(TM) Management Extensions (JMX TM) Reference Implementation v1.2. <http://java.sun.com/products/JavaManagement/download.html>

Using Previews to Reduce the Cost of Video-on-Demand Services

Jehan-François Pâris¹

Department of Computer Science, University of Houston,
Houston, TX 77204-3010
paris@cs.uh.edu

Abstract. We propose to start each video-on-demand program by two or three short previews of coming attractions as it is customarily done in movie theaters and on videocassettes. These previews will be cheap to distribute for they can be viewed in any order. As we will show, they can significantly reduce the bandwidth requirements of most video distribution protocols, among which, stream tapping and most static and dynamic video broadcasting protocols. The sole exception seems to be fixed-delay broadcasting protocols that limit the receiving bandwidth of the customers' set-top boxes to two channels. In that case, the bandwidth used for distributing the previews temporarily reduces the bandwidth that remains available to the fixed-delay broadcasting protocol, which results in a significant increase of the server bandwidth required to achieve a given customer waiting time.

1 Introduction

When we go watch a movie in a theater, when we put a videocassette in our VCR, we are to watch first a few short previews of coming attractions. The main purpose of these previews is to entice us either to come back to the theater for another movie or to rent more videocassettes.

We think that video-on-demand (VOD) services should follow the same approach and start their programs by two to three short previews. First, it would allow them to advertise their current and future offerings. Second, it could significantly reduce the bandwidth requirements of the service.

This is an important issue because distributing videos on demand necessitates vast amounts of bandwidth and these high bandwidth requirements are the main reason for the very high cost of VOD services. Reducing this cost is an imperative because VOD has to compete against cheaper well-established rivals such as videocassette rentals or pay-per-view.

¹ Supported in part by the National Science Foundation under grant CCR-9988390

This situation has resulted in numerous proposals aiming at reducing the bandwidth requirements of VOD services. All these proposals can be broadly classified into three main categories. All proposals in the first group follow a strict *proactive* approach. They partition each video into *segments* and retransmit these segments according to a fixed schedule guaranteeing that any customer having waited for a given maximum delay w will be able to watch the whole video without any interruption. The simplest proactive protocol is *staggered broadcasting*. It consists of broadcasting each video on k distinct channels with each broadcast starting an offset D/k , where D is the duration of the video. More sophisticated proactive protocols, such as *pyramid broadcasting* [17], can achieve shorter customer waiting times at much lower bandwidth costs. These protocols require customer set-top boxes that can receive video data at at least twice the video consumption rate and have enough local storage to store up to one half of the video duration.

Proactive protocols work well for videos that are in high demand such as the ten to twenty videos that are "hot" at any time. Reactive protocols take a less radical approach and do not require the video server to transmit any data in the absence of any request for the video. They try however to share as many data as possible among overlapping requests for the same video. Some of the earliest reactive solutions include *batching* [5] and *piggybacking* [8]. More recent proposals such as *stream tapping* [2, 3] or *hierarchical multicast stream merging* [6] save more bandwidth but also require customer set-top boxes that can receive video data at at least twice the video consumption rate and have enough local storage to store at least ten to twenty minutes of video.

A third group of solutions are reactive in nature but behave like proactive solutions once the customer request arrival rate reaches some saturation point [11, 18]. They work best for videos whose popularity varies over time.

As we will see, all three approaches can benefit from starting each video program with a few minutes of previews. Previews are cheap to distribute for they can be viewed in any order. They also provide the customer with something interesting to watch while waiting for the beginning of the video. This would allow us to stretch this delay and reduce the bandwidth requirements of nearly every video distribution protocol. We need however to qualify this statement: some broadcasting protocols require the customer set-top box to start accumulating video data while the customer is still waiting for the video. This is the case for the GEBB protocol [9] and the fixed-delay pagoda broadcasting protocol [15]. Whenever the receiving bandwidth of the customer set-top boxes is limited to two channels, the bandwidth used for distributing the previews will temporarily reduce the bandwidth that remains available to the protocol. This will result in a significant increase of the server bandwidth required to achieve a given customer waiting time.

The remainder of the paper is organized as follows. Section 2 reviews relevant previous work on video distribution protocols. Section 3 shows how to use previews to mask the latency of reactive protocols and reduce the bandwidth requirements of proactive protocols. Section 4 compares our approach with partial preloading. Finally, Section 5 has our conclusions.

Slots	1	2	3	4
First Channel	S_1	S_1	S_1	S_1
Second Channel	S_2	S_3	S_2	S_3
Third Channel	S_4	S_5	S_6	S_7

Figure 1. The first three channels for the fast broadcasting protocol

2 Previous Work

We will only mention those protocols that are the most relevant to our work. The reader interested in a more comprehensive review of proactive protocols may want to consult reference [4].

Proactive protocols

The simplest proactive protocol is Juhn and Tseng's *fast broadcasting* (FB) protocol [12]. It allocates to each video k channels whose bandwidths are all equal to the video consumption rate b . It then partitions each video into $2^k - 1$ segments, S_1 to S_{2^k-1} , of equal duration d .

As Figure 1 indicates, the first channel continuously rebroadcasts segment S_1 , the second channel transmits segments S_2 and S_3 , and the third channel transmits segments S_4 to S_7 . More generally, channel j with $1 \leq j \leq k$ transmits segments $S_{2^{j-1}}$ to S_{2^j-1} .

When customers want to watch a video, they wait until the beginning of the next transmission of segment S_1 . They then start watching that segment while their STB starts downloading data from all other channels. Hence the maximum customer waiting time is equal to the duration of a segment. Define a *slot* as a time interval equal to the duration of a segment. To prove the correctness of the FB protocol, we need only to observe that each segment S_i with $1 \leq i \leq 2^k - 1$ is rebroadcast at least once every i slot. Then any client STB starting to receive data from all broadcasting channels will always receive all segments on time.

The FB protocol does not require customer STBs to wait for any minimum amount of time. As a result, there is no point in requiring customer STBs to start downloading data while customers are still waiting for the beginning of the video. The newer *fixed-delay pagoda broadcasting* (FDPB) protocol [15] requires all users to wait for a fixed delay w before watching the video they have selected. This waiting time is normally a multiple m of the segment duration d . As a result, the FDPB protocol can partition each video into much smaller segments than FB with the same number of channels. Since these smaller segments can be packed much more efficiently into the k channels assigned to the video, the FDPB protocol achieves smaller customer waiting times than an FB protocol with the same number of channels.

Figure 2 displays the segment-to-channel mappings of a FDPB protocol requiring customers to wait for exactly 9 times the duration of a segment. Given that delay, the first

Subchannel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 st	S ₁			S ₂			S ₃			S ₄			S ₅		
2 nd		S ₄			S ₅			S ₆			S ₇			S ₈	
3 rd			S ₈			S ₉			S ₁₀			S ₁₁			S ₁₂

(a) First channel

Channel	Subchannels	First Segment	Last Segment
C ₂	All 5 subchannels	S ₁₃	S ₄₂
C ₃	All 7 subchannels	S ₄₃	S ₁₁₆
C ₄	All 11 subchannels	S ₁₁₇	S ₃₀₈
C ₅	All 18 subchannels	S ₃₀₉	S ₈₁₄

(b) Channels C₂ to C₅**Figure 2.** How the fixed-delay broadcasting protocol maps its first five channels.

segment of the video will need to be broadcast at least once every 9 slots. The protocol will use time division multiplexing to partition the first channel into $\sqrt{9}$ subchannels with each subchannel containing one third of the slots of the channel. The first subchannel will continuously broadcast segments S_1 to S_3 ensuring that these segments are repeated exactly once every 9 slots.

Observe that the next segment to be broadcast, segment S_4 needs to be broadcast once every 12 slots. Hence the second subchannel will transmit segments S_4 to S_7 ensuring that these segments are repeated exactly once every 12 slots. In the same way, the third subchannel will broadcast segments S_8 to S_{12} ensuring that these segments are repeated exactly once every 15 slots.

The process will be repeated for each of the following channels partitioning each channel into a number of subchannels equal to the square root of the minimum periodicity of the lowest numbered segment to be broadcast by the channel. Hence channel C_2 will be partitioned into 5 subchannels because segment S_{13} needs to be repeated every 21 slots and $5 = \sqrt{21}$. As a result, the protocol will map segments S_{13} to S_{42} into the 5 subchannels of the second channel. Applying the same process on channels C_3 to C_5 , the protocol will be able to map 814 segments into five channels and achieve a deterministic waiting time of 9/814 of the duration of the video, that is, 80 seconds for a two-hour video.

Reactive protocols

Stream tapping [2, 3] or *patching* [10], assumes that each customer STB has a buffer capable of storing at least 10 minutes of video data. This buffer will allow the STB to

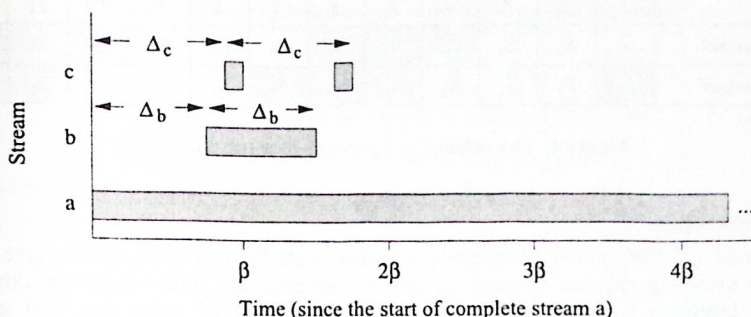


Figure 3. Stream tapping.

"tap" into streams of data on the VOD server originally created for other clients, and then store these data until they are needed. In the best case, clients can use most of the data from the existing stream, and greatly reduce the amount of time they need their own stream.

Stream tapping defines three types of streams. *Complete streams* read out a video in its entirety. These are the streams clients typically tap from. *Full tap streams* can be used if a complete stream for the same video started $\Delta \leq \beta$ minutes in the past, where β is the size of the client buffer, measured in minutes of video data. In this case, the client can begin receiving the complete stream right away, from its buffer, which will then always contain a moving Δ -minute window of the video. Stream tapping also defines partial tap streams, which can be used when $\Delta > \beta$. In this case clients must go through cycles of filling up and then emptying their buffer since the buffer is not large enough to account for the complete difference in video position. To use tap streams, clients only have to receive at most two streams at any one time. If they can actually handle a higher bandwidth than this, they can use an option to the protocol called extra tapping. Extra tapping allows clients to tap data from any stream on the VOD server, and not just from complete streams. Figure 3 shows some example streams from the VOD server's perspective. Stream *a* is a complete stream, and it must exist for the entirety of the video. Stream *b* is a full tap stream starting Δ_b minutes after stream *a*. It only has to exist for Δ_b minutes. Stream *c* is another full tap stream, but it is able to use extra tapping to tap data from stream *b*, and so its service time is much smaller than Δ_c .

Eager, Vernon and Zahorjan's *hierarchical multicast stream merging* (HMSM) protocol [6] is a reactive protocol that never requires the STB to receive more than two streams at the same time. *Selective catching* combines both reactive and proactive approaches. It dedicates a certain number of channels for periodic broadcasts of videos while using the other channels to allow incoming requests to catch up with the current broadcast cycle [7].

Slots	1	2	3	4	5	6	7	8	9	10	11	12
1 st channel	S ₁	S ₂	S ₃	S ₁	S ₂	S ₃	S ₁	S ₂	S ₃	S ₁	S ₂	S ₃
2 nd channel	S ₁	S ₁	S ₁	S ₂	S ₁	S ₁	S ₂	S ₁	S ₁	S ₂	S ₁	S ₁

Figure 4. Broadcasting two previews over two channels

Other techniques

Basu *et al.* [1] have proposed inserting advertisements in video programs to give more time to incoming requests to catch up with earlier requests. Their technique does not require customer STBs to buffer any significant amount of video data and greatly improves the performance of piggybacking protocols [8].

Partial preloading [13–14, 16] loads in the customer STB the first few minutes of the top 10 to 20 videos in order to provide zero-delay access to these videos and reduce the server bandwidth required by the remainder of the video. It applies to both proactive and reactive distribution protocols.

3 Implementation

We discuss first how previews can be distributed and show then how to use previews to mask the latency of a fixed-delay broadcasting protocols and reduce the bandwidth requirements of stream taping protocols.

Distributing previews

We will focus here on the simplest case where all customers watch the same previews. More targeted strategies are possible but they would also require more server bandwidth.

The cheapest way to distribute the previews is to broadcast them in sequence on a single channel. Since previews can be watched in any order, the average customer waiting time will be equal to one half of the average duration of a preview. A much better quality of service could be achieved by broadcasting the previews on two channels. As seen on Figure 4, adding an extra channel would allow us to partition each preview into three segments and achieve an average customer waiting time equal to one sixth of the average duration of a preview. Since most previews last around two minutes, this means an average waiting time of 20 seconds (and a maximum waiting time of 40 seconds).

Using previews to mask the latency of a fixed-delay broadcasting protocol

Fixed-delay broadcasting protocols, such as the GEBB protocol [9] and the FDPB protocol [15], could greatly benefit from our approach. Consider for instance the case of a FDPB protocol with $m = 9$. As we saw earlier, it requires 5 channels to achieve a waiting time of 40 seconds for a two-hour video. Assume now that each video is preceded by 4 minutes of previews. As shown on Figure 2, 4 channels suffice to partition the video into

<i>Channel</i>	<i>Number of Subchannels</i>	<i>First Segment</i>	<i>Last Segment</i>
C_2	3	S_1	S_{12}
C_2	5s	S_{13}	S_{42}
C_3	6	S_{43}	S_{95}
C_4	8	S_{96}	S_{193}
C_5	11	S_{193}	S_{369}
C_6	14	S_{370}	S_{674}

Figure 5. How the FDPB protocol maps its first six channels when the STB receiving bandwidth is limited to two channels.

308 segments and achieve a waiting time of $9/308$ of the video duration. This waiting time would be completely masked by the 4 minutes of previews as long as the video lasts less than $4 \times 308/9 = 137$ minutes. Increasing the duration of the previews to five minutes would allow us to mask the latency of videos lasting up to $5 \times 308/9 = 171$ minutes, that is, almost three hours.

Hence we could reduce the bandwidth requirements of the FDPB by 20 percent by having customers watch between four and five minutes of previews before each video.

Handling set-top boxes that cannot receive more than two channels at the same time

Like most other broadcasting protocols, the FDPB protocol assumes that the STB can and will simultaneously receive data from all the channels on which the various segments of the video are broadcast. This requirement complicates the design of the STB and increases its cost.

Fortunately, the FDPB protocol can be easily modified to handle customers connected to the video-on-demand service through a STB that cannot receive data at more than twice the video consumption rate [15]. Consider the case of a FDPB protocol with $m = 9$ that interacts with STBs that cannot receive video data from more than two channels. As shown in Figure 5, the segment-to-channel mappings of the first two channels are unchanged. The first mappings to be affected are those of channel C_3 as the STB must now wait until it has received all data from the first channel before starting to receive data from channel C_3 .

The last segment broadcast by the first channel is segment S_{12} . As shown on Figure 2, it is broadcast once every 15 slots. The first segment broadcast by channel C_3 is segment S_{43} . Recalling that the customer waiting time is equal to 9 slots, we see that segment S_{43} must now be broadcast at least once every $43 + 8 - 15 = 36$ slots. Similarly segment S_{44} has now to be broadcast at least once every $44 + 8 - 15 = 37$ slots and so on. As a result, channel C_3 will now be partitioned into $\sqrt{36} = 6$ subchannels and will broadcast segments S_{43} to S_{95} . The process will be repeated on the remaining channels subject to the

Channel	Number of Subchannels	First Segment	Last Segment
C_2	3	S_1	S_{12}
C_2	3	S_{13}	S_{28}
C_3	5	S_{29}	S_{58}
C_4	6	S_{59}	S_{112}
C_5	9	S_{113}	S_{213}
C_6	12s	S_{214}	S_{391}

Figure 6. How the FDPB protocol maps its first six channels when the STB receiving bandwidth is limited to two channels and the customer watches previews while waiting for the video.

restriction that the STB cannot receive data from channel C_j for $j \geq 3$ until it has finished receiving data from channel C_{j-2} .

As we can see from Figure 5, the FDPB protocol will now need 6 channels to achieve a waiting time of 9/674 of the video duration, that is, 96 seconds for the same two-hour video.

Assume now that we want to mask this waiting time by having the customer watch one short preview. Since this preview will occupy one of the two channels the STB can listen to at any given time, the STB will not be able to receive data from channel C_2 until after the STB has finished receiving the previews. In our case, this means that the first segment broadcast by channel C_2 , that is, segment S_{13} will now have to be repeated at least once every 12 slots instead of at least once every $12 + 8 = 20$ slots. Consequently, channel C_2 will only be able to broadcast segments S_{13} to S_{28} , that is, 15 segments instead of the 30 segments it was previously broadcasting. This will result in much less efficient mappings for the each and every channel for all segments will now have to be repeated more frequently than before.

As we can see on Figure 6, having customers watching previews while waiting for the video of their choice has a dramatic impact on the performance of the FDPB protocol, which can only pack now 391 segments into six channels. As a result, the customer waiting time for a two-hour video will be equal to $7200 \times 9/391 = 203$ seconds.

We can thus characterize the benefits of having customers watching previews while waiting for the video of their choice as rather dubious. First and foremost, we will not save any bandwidth. Second, it is not clear that the customer will prefer waiting for almost three minutes while watching previews to waiting for slightly more than one minute and half in front of a blank screen.

Using previews to reduce the latency of a stream tapping protocol

Stream tapping and hierarchical multicast stream merging have two major advantages over broadcasting protocols. First, they provide true instant access to the video. Second,

they require much less bandwidth than broadcasting protocols to distribute videos that are not requested more than ten times per hour for a two-hour video.

Unfortunately, the same is not true at higher request arrival rates. Since they handle all customer requests one by one, proactive protocols require much more bandwidth than most broadcasting protocols whenever the request arrival rate exceeds 20 requests per hour. One attractive solution would be to start each video program by four to five minutes of previews. This would have the same effect as batching requests together over a fixed batching interval equal to the duration of these previews and would greatly reduce the number of video streams that the server has to manage.

To estimate the impact of these previews on the protocol bandwidth requirements, we can observe that starting each video program by D_p minutes of previews is equivalent to limit the customer arrival rate to at most one arrival each D_p minutes. Previous simulation studies have shown that stream tapping with all its bandwidth savings options activated requires an average of 5.55 channels to satisfy 20 requests per hour for a two-hour video [14]. Hence starting each video with as little as three minutes of previews should allow us to satisfy any number of requests with an average bandwidth of 5.5 channels.

Increasing the durations of the previews to five to six minutes would result in further bandwidth reductions. For instance, starting each video with six minutes of previews would reduce the average bandwidth required to distribute a video to 4.09 channels, a figure quite similar to the bandwidth requirements of a FDPB protocol starting each video program with four minutes of previews.

Using previews with hybrid distribution protocols

These protocols, also known as dynamic broadcasting protocols, are reactive in nature but behave like proactive protocols once the customer request arrival rate reaches some saturation point [11, 18]. Hence, their bandwidth requirements are bounded, which distinguishes them from purely reactive protocols.

This distinction becomes blurred once we introduce previews. As we have just seen, adding previews to video programs puts an upper bound on the bandwidth requirements of a purely reactive protocol such as stream tapping. As a result, one of the motivations for using hybrid distribution protocols will disappear. We should focus instead on the actual performance of hybrid protocols all over the range of request arrival rates. An efficient hybrid protocol, such as the channel-based heuristic distribution protocol [18], would still perform as well as a stream tapping protocols at low request arrival rates while requiring significantly less bandwidth at high request arrival rates.

4 Comparison with Partial Preloading

Starting each video program by a few minutes of previews can have a dramatic impact on the bandwidth requirements of most video distribution protocols. In fact, it compares favorably with partial preloading [13–14, 16]. Starting each video program with x minutes of previews provides nearly the same results as having the first x minutes of each video preloaded in the customer STB.

Consider, for instance the case of the FDPB protocol. We have seen that having customers watch between four and five minutes of previews before each video would allow us to broadcast videos on four channels instead of five. This is not very different from what we could achieve by partitioning each video into 317 segments and by having the first 9 segments preloaded in the customer STB. As Figure 2 shows, the remaining 308 segments could then be broadcast on 4 channels. In both cases, we would have to factor the cost of the one or two additional channels required to broadcast either the previews or the initial segments of each video but the cost of these additional channels could be shared among all the videos being broadcast by the server.

There are however some important differences between the two techniques. First, prepending previews is much easier to implement for it does not require any negotiation with the customer STB. Second, applies to all videos while partial preloading only applies to a limited number of presumed popular videos.

On the other hand, partial preloading does not force customers to watch a few minutes of previews before watching the video of their choice. Hence the amount of previews we can force the customer to watch is quite limited.

Combining both techniques could be an attractive proposition. We do not believe it would make much sense to preload previews in the customers' STBs for it would not lead into any significant bandwidth improvements. A more attractive option would be to make partial preloading optional and have customers who do not have the first few minutes of the video preloaded on their STB watch more previews than those who have them.

5 Conclusion

Starting each video program by four to five minutes of previews would only require one to two additional video channels and could significantly reduce the bandwidth requirements of most video distribution protocols. In particular, it would reduce by 20 percent the bandwidth requirement of a fixed-delay pagoda protocol and would put an upper bound on the bandwidth requirements of stream taping protocols. Fixed-delay broadcasting protocols that limit the receiving bandwidth of the customer set-top boxes to two channels seem to be the sole exception. In their case, the bandwidth used for distributing the previews temporarily reduces the bandwidth that remains available to the fixed-delay broadcasting protocol, which results in a 20 to 25 percent increase of the server bandwidth required to achieve a given customer waiting time.

References

1. Basu, P., A. Narayanan, W. Ke, T. D. C. Little, and A. Bestavros. Optimal scheduling of secondary content for aggregation in video-on-demand systems. *Proc. 8th Int'l Conf. on Computer Communications and Networks*, pp. 104–109, Oct. 1999.

2. Carter, S. W. and D. D. E. Long. Improving video-on-demand server efficiency through stream tapping. *Proc. 6th Int'l Conf. on Computer Communications and Networks*, pp. 200–207, Sep. 1997.
3. Carter, S. W. and Darrell D. E. Long. Improving bandwidth efficiency of video-on-demand servers. *Computer Networks*, 31(1-2):99–111, 1999.
4. Carter, S. W., D. D. E. Long and J.-F. Pâris. Video-on-demand broadcasting protocols. In *Multimedia Communications: Directions and Innovations* (J. D. Gibson, Ed.), Academic Press, San Diego, pp. 179–189, 2000.
5. Dan, A., D. Sitaram, and P. Shahabuddin. Dynamic batching policies for an on-demand video server. *ACM Multimedia Systems Journal*, 4(3):112–121, June 1996.
6. Eager, D. L., M. K. Vernon and J. Zahorjan. Minimizing bandwidth requirements for on-demand data delivery. *IEEE Trans. on Knowledge and Data Engineering*, 13(5):742–757, Sept./Oct. 2001.
7. Gao, L., Z.-L. Zhang and D. Towsley. Catching and selective catching: efficient latency reduction techniques for delivering continuous multimedia streams. *Proc. 1999 ACM Multimedia Conf.*, pp. 203–206, Nov. 1999.
8. Golubchik, L., J. Lui, and R. Muntz. Adaptive piggybacking: a novel technique for data sharing in video-on-demand storage servers. *ACM Multimedia Systems Journal*, 4(3): 140–155, 1996.
9. Hu, A., I. Nikolaidis, and P. van Beek. On the design of efficient video-on-demand broadcast schedules. *Proc. 7th Int'l Symp. on Modeling, Analysis, and Simulation of Computing and Telecommunication Systems*, pp. 262–269, Oct. 1999.
10. Hua, K. A., Y. Cai, and S. Sheu. Patching: a multicast technique for true video-on-demand services. *Proc. 1998 ACM Multimedia Conf.*, pp. 191–200, Sep. 1998.
11. Hua, K. A., J. H. Oh, and K. Vu. An adaptive hybrid technique for video multicast. *Proc. 7th IEEE Int'l Conf. on Computer Communications and Networks*, pp. 227–234, Oct. 1998.
12. Juhn, L. and L. Tseng. Fast data broadcasting and receiving scheme for popular video service. *IEEE Trans. on Broadcasting*, 44(1):100–105, March 1998.
13. Pâris, J.-F., D. D. E. Long and P. E. Mantey. A zero-delay broadcasting protocol for video on demand. *Proc. 1999 ACM Multimedia Conf.*, pp. 189–197, Nov. 1999.
14. Pâris, J.-F. A stream tapping protocol with partial preloading. *Proc. 9th Int'l Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 423–430, Aug. 2001.
15. Pâris, J.-F. A fixed-delay broadcasting protocol for video-on-demand. *Proc. 10th Int'l Conf. on Computer Communications and Networks*, pp. 418–423, Oct. 2001.
16. Sul, H.-K., H.-C. Kim, and K. Chon. A hybrid pagoda broadcasting protocol: fixed-delay pagoda broadcasting protocol with partial preloading. *Proc. 2003 IEEE Int'l Conf. on Multimedia and Expo*, July 2003.
17. Viswanathan, S. and T. Imielinski. Metropolitan area video-on-demand service using pyramid broadcasting. *ACM Multimedia Systems Journal*, 4(4):197–208, 1996.
18. Zhang, Q. and J.-F. Pâris. A channel-based heuristic distribution protocol for video-on-demand. *Proc. 2002 IEEE Int'l Conf. on Multimedia and Expo*, Vol. 1, pp. 245–248, Aug. 2002.

Extensiones gramaticales del álgebra relacional para el soporte de integridad en sistemas relacionales

Beatriz Bernábe Loranca¹, Ma. del Rocío Boone Rojas¹, Ramiro López Sales²

¹ Área de Bases de Datos

Facultad de Ciencias de la Computación, BUAP

Calle 14 sur y Avenida San Claudio, Colonia San Manuel, Puebla, México

rboone@cs.buap.mx, beti@cs.buap.mx

www.cs.buap.mx/~bety/Investigacion.html

²Facultad de Ciencias de la Computación, BUAP

Calle 14 sur y Avenida San Claudio, Colonia San Manuel, Puebla, México

ramiro@cs.buap.mx

Resumen. Actualmente la gran mayoría de los Sistemas Administradores de B.D. Relacionales ofrecen un soporte limitado para la representación y comprobación de restricciones de integridad, de tal manera que es necesario proporcionar algunas alternativas de solución para esta situación.

Con el propósito de ofrecer un conjunto de reglas gramaticales que permitan la definición y el tratamiento de restricciones de integridad a nivel de un DBMS, en este trabajo se presenta una gramática extendida basada en un intérprete para el álgebra relacional, en un lenguaje de integridad y en especificaciones complementarias para la regla de integridad referencial del modelo relacional.

1 Introducción

Para asegurar que nuestra información dentro de una BD sea correcta y por ende de calidad, debemos de garantizar que nuestro DBMS proporcione un soporte satisfactorio para la representación y comprobación de restricciones de integridad. El modelo relacional y por ende los DBMS basados en él, contemplan restricciones de integridad generales asociadas a las llaves primarias y a las llaves ajenas, actualmente algunos DBMS soportan estas restricciones de integridad. Sin embargo la gran mayoría se encuentran lejos de proporcionar un soporte satisfactorio para el tratamiento de integridad, como lo hemos mostrado en los trabajos [5,9]. De tal forma que se deben plantear alternativas de solución para esta situación.

El problema de integridad en bases de datos, se identifica como uno de los problemas clásicos en Bases de Datos, cuya investigación ha dado lugar a diferentes contribuciones para su tratamiento tanto en el nivel lógico, como es el caso de las especificaciones complementarias para el componente de integridad del modelo relacional y el lenguaje de integridad propuestos en [2, 3], como en el nivel conceptual, como es el caso del análisis y extensión del modelo entidad relación, que se presenta en [11].

Por otra parte, se ha analizado en [10] la pertinencia de dar tratamiento a las restricciones de integridad a nivel del DBMS, de tal manera que sea nuestro DBMS el que se encargue de verificar automáticamente las restricciones cada vez que se actualicen los datos.

Dentro del marco del tratamiento de integridad a nivel lógico de una base de datos y con el propósito de ofrecer un conjunto de reglas gramaticales que permitan la definición y el tratamiento de restricciones de integridad en principio, bajo control del DBMS, en este trabajo se presenta una gramática extendida basada en referencias tales como un intérprete para el álgebra relacional[7], un lenguaje de integridad [2,3] y en especificaciones complementarias para la regla de integridad referencial del modelo relacional [2,3].

El trabajo se organiza como sigue: en la sección dos se incluyen aspectos fundamentales del concepto de integridad y del componente de integridad del modelo relacional, la sección tres proporciona las especificaciones complementarias que se han propuesto para la regla de integridad referencial, la siguiente sección presenta una breve descripción de un lenguaje de integridad, mediante el cual, en particular se formulan las reglas de integridad relacionales, en la sección cinco se describen las ideas básicas relacionadas con el intérprete del álgebra relacional que proporciona la base para el presente trabajo, en la sección seis, se presentan las extensiones de la gramática desarrollada para el intérprete del álgebra relacional. Finalmente se presentan las conclusiones y perspectivas de este trabajo.

2 El concepto de integridad

El término integridad se refiere a la exactitud o corrección y últimamente a la calidad de los datos en la base de datos, para ello es necesario incluir ciertas *reglas de integridad*, cuyo propósito es informar al DBMS de ciertas restricciones del mundo real (como la restricción de que los pesos de los artículos no pueden ser negativos) para que pueda impedir la ocurrencia de tales configuraciones imposibles de valores. Ullman [4], establece una clasificación general de las restricciones de uso común:

- a) Las llaves son atributos o conjuntos de atributos que proporcionan la identificación única de un objeto dentro de su clase o de una entidad dentro de su conjunto entidad.
- b) Las restricciones de valor único son requisitos de que el valor de un papel determinado sea único.
- c) Las restricciones de integridad referencial son requisitos de que el valor referenciado por algún objeto exista realmente en la base de datos.
- d) Las restricciones de dominio requieren que el valor de un atributo sea extraído de un conjunto específico de valores o bien que se encuentre dentro de un intervalo determinado.
- e) Las restricciones generales son afirmaciones arbitrarias que deben cumplirse en la base de datos.

En el caso del modelo relacional, el componente de integridad, incluye restricciones asociadas a las llaves primarias y ajenas respectivamente, que en términos genera-

les establecen que las llaves primarias de las relaciones base no deben contener nulos y que la base de datos no debe contener valores de llave ajena sin concordancia. Consecuentemente, los DBMS que se identifican como relacionales, deberían de ofrecer un soporte con el alcance suficiente como para dar un tratamiento adecuado a las restricciones de integridad que se pueden derivar a partir de las reglas de integridad del modelo relacional.

2.1 Reglas de integridad del modelo relacional

El componente de integridad del modelo relacional, ha permitido realizar algunas propuestas de especificación complementarias que permiten en la práctica utilizar y dar soporte a las reglas de integridad del modelo relacional, como es el caso de los trabajos de Date [2] que se citan en el presente trabajo. El modelo relacional incluye dos reglas generales de integridad, estas dos se refieren a las claves primarias y a las claves ajenas, las cuales Date [2, 3], las define a partir de los siguientes conceptos.

Una llave candidata se define de la siguiente forma. El atributo K (posiblemente compuesto) de la relación R es una llave candidata de R si y sólo si satisface las dos propiedades, independientes del tiempo: 1) Unicidad: Jamás, ningún valor válido de R contiene dos tuplas distintas con el mismo valor de K. 2) Irreducibilidad: Ningún subconjunto propio de K tiene la propiedad de unicidad.

Cabe señalar que por la propiedad de las relacionales que establece que las relaciones no contienen tuplas repetidas, toda relación tiene por lo menos una llave candidata. En la práctica, las relaciones tienden a tener una y sólo una llave candidata, pero indudablemente, es posible que tengan más. Por otro lado, del conjunto de llaves candidatas de una relación dada, se elige una y sólo una como llave primaria de esa relación, las demás, si existen, se llamarán llaves alternativas.

En términos de los conceptos anteriores se puede enunciar la primera regla de integridad de entidades: "Ningún componente de la llave primaria de una relación base puede aceptar nulos". Con "nulos" se indica información faltante por alguna razón, por ejemplo, si la propiedad no es aplicable o si el valor se desconoce etc. A veces, la regla de integridad de las entidades se expresa como: "En una base de datos relacional, nunca registraremos información acerca de algo que no podamos identificar."

De tal forma que un DBMS relacional que proporcione soporte para esta regla, deberá de verificar que los valores de llave primaria que se intenten introducir mediante las operaciones de inserción y de actualización no sean nulos y que no estén repetidos.

La segunda regla de integridad se aplica a las llaves ajenas; en términos informales, se puede definir una llave ajena de la forma siguiente: Una llave ajena es un atributo (quizá compuesto) de una relación R2 cuyos valores deben concordar con los de la clave primaria de alguna relación R1 (donde R1 y R2 no necesariamente son distintos). Un valor de clave ajena representa una referencia a la tupla donde se encuentra el valor correspondiente a la clave primaria (la tupla referida o tupla objetivo). Por tanto, el problema de garantizar que la base de datos no incluya valores no válidos de una llave ajena se conoce como el problema de integridad referencial.

La restricción según la cual los valores de una llave ajena determinada deben concordar con los valores de la llave primaria correspondiente se conoce como restricción referencial y la relación que contiene a la llave ajena se conoce como relación referencial y la relación que contiene a la clave primaria correspondiente se denomina relación referida o relación objetivo. En términos más precisos, se puede establecer la definición de una llave ajena. El atributo LF (quizá compuesto) de la relación base R2 es una clave ajena si y solo si satisface estas dos propiedades independientes del tiempo: Cada valor de LF es nulo del todo o bien no nulo del todo (con "nulo del todo" o "no nulo del todo" se quiere decir que, si LF es compuesto, todos sus componentes son nulos o bien todos sus componentes son no nulos, no una combinación.). Existe una relación base R1 con clave primaria LP tal que cada valor no nulo de LF es idéntico al valor de LP en alguna tupla de R1.

A partir de los conceptos anteriores se expresa la segunda regla de integridad del modelo relacional, la Regla de Integridad Referencial. "La Base de Datos no debe contener valores de llave ajena sin concordancia". Con el término "valores de llave ajena sin concordancia" se quiere decir aquí un valor no nulo de llave ajena para el cual no existe un valor concordante de la llave primaria en la relación objetivo pertinente. Dicho de otra manera la regla de integridad referencial dice tan solo que si B hace referencia a A, entonces A debe de existir.

3 Especificaciones complementarias para la regla de integridad referencial

De acuerdo al análisis que realiza Date [2], la regla de integridad referencial, requiere de especificaciones complementarias que permitan dar un tratamiento a la misma, que incluya por lo menos las situaciones más comunes que se presentan en la práctica. Podemos citar algunos aspectos del análisis anterior.

La regla de integridad referencial se formula en términos de estados de la base de datos. Cualquier estado de la base de datos que no satisfaga la regla será incorrecto por definición; pero, ¿cómo pueden evitarse tales estados incorrectos? La regla en sí no lo dice con exactitud. Una posibilidad, desde luego, es que el sistema rechazara cualquier operación que en caso de ejecutarse, produciría un estado ilegal. Pero en muchos casos una alternativa preferible sería que el sistema aceptara la operación pero realizara (en caso necesario) ciertas operaciones de compensación con objeto de garantizar un estado legal como resultado final.

Por tanto, para cualquier base de datos, el diseñador de la base de datos deberá de poder especificar cuáles operaciones han de rechazarse y cuáles han de aceptarse, y, en el caso de estas últimas, cuáles operaciones de compensación (si acaso) deberá de realizar el sistema. La idea fundamental es la siguiente. Para cada llave ajena, es necesario responder tres preguntas: ¿Puede aceptar nulos esa llave ajena?, ¿Qué ocurre si se intenta eliminar el objetivo de una referencia de clave ajena? Para responder esta pregunta, existen por lo menos tres posibilidades: Restringir (restricted): La operación de eliminación está "restringida" al caso en el cual no existen tuplas referenciales. Propagar (cascades): La operación de eliminación "se propaga" ("en cascada") y

elimina las tuplas referenciadas. Anular (nullifies): Se asignan nulos a la llave ajena en todas las tuplas referenciadas y en seguida se elimina el objetivo (sólo si la llave ajena puede aceptar nulos). ¿Qué deberá suceder si hay un intento de modificar la llave primaria del objetivo de una referencia de llave ajena? Nuevamente como en el caso anterior, existen por lo menos las mismas tres posibilidades: Restringir (restricted): La operación de modificación está "restringida" al caso en el cual no existen tuplas referenciales. Propagar (cascades): La operación de modificación "se propaga" ("en cascada") y modifica las tuplas referenciadas. Anular (nullifies): Se asignan nulos a la llave ajena en todas las tuplas referenciadas y en seguida se modifica el objetivo (sólo si la llave ajena puede aceptar nulos).

Para dar un tratamiento a los casos más comunes que pueden presentarse en la práctica para la regla de integridad referencial, basada en la propuesta de Date [2], proponemos como en [8], la siguiente sintaxis declarativa para una cláusula FOREIGN KEY (sintaxis de llave ajena):

```
FOREIGN KEY (clave ajena) REFERENCES objetivo
    NULLS [not] ALLOWED
    DELETE OF objetivo efecto (1.1)
    UPDATE OF clave-primaria-del-objetivo efecto
```

donde el 'efecto' es RESTRICTED, CASCADES, NULLIFIES(SET NULL) o Reg_Int_Part, que es el identificador de un procedimiento que puede definir el usuario. La cual permite al diseñador de la B.D. especificar no solo el atributo o combinación de atributos que constituye una llave ajena y la relación objetivo a la cual hace referencia esa clave ajena, sino también las reglas particulares para claves ajenas —las reglas de nulos, eliminación y modificación— aplicables a esa llave ajena. O en general, satisfacer las necesidades particulares de diseñadores de bases de datos, al considerar que "efecto" en la sintaxis anterior incluya la posibilidad de invocar un procedimiento de base de datos para dar el tratamiento adecuado a la regla de integridad requerida por el diseñador.

4 Lenguaje de integridad hipotético

Date [2], ha propuesto un lenguaje hipotético para la formulación de reglas de integridad, que permitan especificar aspectos de su comportamiento. A continuación se incluyen algunos de sus aspectos y ejemplos asociados. El Lenguaje de integridad que se propone incluye un par de proposiciones, CREATE INTEGRITY RULE (crear regla de integridad) y DROP INTEGRITY RULE (desechar regla de integridad). En general, la proposición CREATE INTEGRITY RULE (crear regla de integridad) debe especificar cuatro componentes:

1) El nombre de la regla, 2) Una o más ocasiones de verificación, 3) Una restricción que debe ser satisfecha para los estados legales de la base de datos, 4) Una respuesta de violación, para indicar qué debe hacerse si la verificación falla.

Por ejemplo, se cita el caso de la regla R1.

```
CREATE INTEGRITY RULE R1
    ON          INSERT S.SITUACIÓN,
               UPDATE S.SITUACIÓN:
```

```
CHECK FORALL S ( S.SITUACIÓN > 0 )
ELSE REJECT;
```

En la sintaxis anterior, se especifica que el nombre de la regla es R1. Las ocasiones de verificación (ON INSERT S.SITUACIÓN, UPDATE S.SITUACIÓN), ("al insertar S.SITUACIÓN, modificar S.SITUACIÓN") en el ejemplo, para indicar al sistema cuándo debe realizar la verificación especificada en la cláusula CHECK (verificar).

La restricción, que debe ser satisfecha por todos los estados legales de la base de datos, ("FORALL S (S.SITUACIÓN > 0)" ("para toda S (S.SITUACION > 0)" en este caso); La respuesta de violación, para indicar qué debe hacerse si la verificación falla ("REJECT" ("rechazar") en el ejemplo, lo cual significa que la proposición INSERT o UPDATE culpable, debe rechazarse con un código de retorno apropiado). Para propósitos ilustrativos, se pueden realizar algunas simplificaciones al lenguaje. Las ocasiones de verificación casi siempre serán obvias; por tanto, supondremos que el DBMS es capaz de determinar por sí mismo la ocasión u ocasiones de verificación si no se especifican de manera explícita. La restricción de la cláusula CHECK casi siempre comenzará con un cuantificador universal. En este caso se omite el cuantificador y se supone que cualquier variable no cuantificada se cuantificará en forma automática con un FORALL (para toda) apropiado. También se supone que, si se omite la cláusula ELSE (si no), quedará implícita la respuesta de violación "rechazar la operación de actualización (con un código de retorno adecuado)".

A través del lenguaje de integridad que se ha citado anteriormente, a continuación se incluyen reglas de integridad relacionadas con las llaves ajenas y primarias de un sistema relacional.

Considere que el campo P# es la llave primaria de la tabla P, de tal forma que las restricciones de llaves primarias se pueden expresar en el lenguaje básico de integridad como:

```
CREATE INTEGRITY RULE INT_ENT
  BEFORE INSERT OF P FROM NUEVO_P.P#
    UPDATE OF P.P# FROM NUEVO_P.P# :
    CHECK NOT (IS_NULL (NUEVO_P.P#) )
    AND NOT EXIST PX (PX.P# = NUEVO_P.P# ) ;
```

Si se considera que el campo P# es una llave ajena en la tabla SP, y concuerda con la llave primaria de la tabla P, de acuerdo a la sintaxis (1.1) para las llaves ajenas, un efecto RESTRICTED, para el ejemplo considerado,

```
DELETE OF P RESTRICTED
UPDATE OF P.P# RESTRICTED
```

las reglas de restricción podrían representarse como:

```
CREATE INTEGRITY RULE INT_REF_REST1
  AFTER INSERT OF SP, UPDATE OF SP.P# :
  CHECK EXIST P (P.P# = SP.P#) ;
CREATE INTEGRITY RULE INT_REF_RES2
  BEFORE DELETE OF P, UPDATE OF P.P# :
  CHECK NOT EXIST SP (SP.P# = P.P#) ;
```


Un efecto CASCADES para el ejemplo considerado,

```
DELETE OF P CASCADES
UPDATE OF P.P# CASCADES
```

la versión "se propaga" de las reglas anteriores se puede representar como:

```
CREATE INTEGRITY RULE INT_REF_CAS1
  BEFORE DELETE OF P:
    CHECK NOT EXIST SP ( SP.P# = P.P#)
    ELSE DELETE SP WHERE SP.P# = P.P#

CREATE INTEGRITY RULE INT_REF_CAS2
  BEFORE UPDATE OF P.P# FROM NUEVO_P.P# :
    CHECK NOT EXIST SP ( SP.P# = P.P#)
    ELSE UPDATE SP.P# FROM NUEVO_P.P#
      WHERE SP.P# = P.P# ;
```

Como se puede observar en estas dos últimas reglas, se ilustra el hecho de que en la parte CHECK ... ELSE ... (verificar ... si no ...) de la proposición CREATE INTEGRITY RULE (crear regla de integridad) representa en realidad un procedimiento disparado, el cual debe invocarse cuando se presenta una condición de disparo especificada. En este caso, el objetivo de esos procedimientos disparados es realizar ciertas acciones de compensación (en caso necesario) para garantizar que la base de datos permanezca en un estado de integridad después de haberse ejecutado el conjunto completo de operaciones.

5 Intérprete de algebra relacional

El componente de manipulación del modelo relacional, está conformado por:

1. Un conjunto de operadores que forman la llamada Álgebra Relacional y,
2. Una operación de asignación que asigna el valor de alguna expresión arbitraria del álgebra a una relación nombrada.

En el álgebra relacional se distinguen las operaciones tradicionales de conjuntos más otras especiales de alto nivel que operan sobre relaciones. Cada uno de estos operadores toma una o dos relaciones como entrada y produce una nueva relación como resultado. Las operaciones tradicionales de conjuntos que se conservan son la unión, intersección, diferencia y producto cartesiano. Y se consideran también, las operaciones relacionales especiales restricción, proyección, reunión y división.

En esencia, dado que el resultado de cualquier operación es un objeto del mismo tipo que los operadores - todos son relaciones - el resultado de una relación puede convertirse en operando de otra, de tal forma que es posible formular expresiones relacionales anidadas.

La gramática libre de contexto original BNF (Backus Naus Form), que se desarrolló en el trabajo [7] genera un pequeño lenguaje para definir estructuras de bases de datos de acuerdo al componente estructural del Modelo Relacional, así como para manipularlas según el álgebra relacional. Este lenguaje está compuesto por un Lenguaje de Definición de Datos (DDL) y un Lenguaje de Manipulación de Datos

(DML). Tanto el DDL como el DML constituyen un intérprete en lenguaje "C". Este intérprete es la herramienta de programación que reconoce un DDL elemental como un subconjunto de instrucciones de un lenguaje anfitrión y que permite definir a las relaciones que se operan. Finalmente el DML manipula algebraicamente las relaciones definidas por el DDL.

Cabe mencionar que el intérprete desarrollado en [7], incluye su analizador sintáctico, su analizador semántico y sus respectivos generadores de errores. Las estructuras definidas por la gramática son totalmente compatibles con estructuras "X-Base" y en conjunto con las herramientas de programación reconocedores de sintaxis y semántica, se logró una exitosa integración con estructuras de CLIPPER, con lo cual podemos afirmar que la gramática extendida que presentamos en la siguiente sección produce reglas gramaticales de integridad compatibles con diferentes productos DBMS relacionales.

6 Gramática para soporte de integridad

Basado en la sintaxis propuesta por Date [2,3], en el lenguaje de integridad hipotético [2,3] considerado en este trabajo, y la gramática para álgebra relacional [7]; en esta sección, se incluye una extensión sintáctica para definir reglas de integridad asociadas a las llaves primarias y ajenas. La extensión de esta gramática no está limitada, dado que es recursiva y ya ha sido probada con productos X-Base, siendo estas las cualidades que invitan a enriquecer el marco gramatical que nos ocupa, y sobre todo, proseguir en el refinamiento de la solución a los problemas de integridad.

La gramática mencionada se describe a continuación, cada producción se acompaña de acciones semánticas que pueden ser abordadas en la codificación o en tiempo de ejecución.

Cabe subrayar que algunas producciones solo se han mencionado ya que exhibirlas explícitamente, llevan a definir otro sub-lenguaje y que precisamente estamos trabajando: $\langle \text{instrucción} \rangle \rightarrow \text{IF} \mid \text{FOR} \mid \dots$

$\mid \langle \text{instr_sql} \rangle \dots, \langle \text{crea_r_rel} \rangle \rightarrow \langle \text{regla_relacional} \rangle \langle \text{regla_entidades} \rangle \dots, \langle \text{condicion} \rangle \rightarrow \langle \text{expresion anidada} \rangle$

Especificaciones gramaticales complementarias para el soporte de integridad

$G = \langle V_n, V_t, P, S \rangle$

Donde: V_n es el conjunto de símbolos no terminales,

V_t es el conjunto de símbolos terminales,

P es el conjunto de producciones y

S es el conjunto que consta del símbolo inicial.

$V_n = \{ \text{prop_alg}, \text{def_alias}, \text{asignación}, \text{alias}, \text{nombre_rel}, \text{list_espec_atr}, \text{exp_alg}, \text{espec_atr}, \text{nombre_atr}, \text{selección}, \text{proyección}, \text{exp_infija}, \text{primitiva}, \text{exp_bool}, \text{exp_alg}, \text{op_infijo}, \text{op_comp}, \text{espec_valor}, \text{comparación}, \text{oper_bool}, \text{opdo_bool}, \text{cte_bool}, \text{cte}, \text{id}, \text{cadena}, \text{alfanum}, \text{letra}, \text{dígito}, \text{cte_alfanum}, \text{cte_num}, \text{número}, \text{num1},$

definición, def_campos, tamaño, defcampo, decim, definiciones, bloque, mini_relacional, define_tablas, define_reglas, define_tabla, define_regla, regla_primaria, regla_ajena, crea_regla, objetivo, efecto, claveprim_objetivo, def_claveprim, regla_integridad, borra_regla, accion1, accion, situacion, cond_ok, cond_nook, expr_bool1, expr_bool2, instrucción, nombre_regla, restriccion, condicion, resp_violacion, id_r, regla_integridad_particular, disparador, regla_primaria, regla_relacional, regla_entidades, crea_r_rel, crea_regla_a, regla_relacional_particular, especificaciones_sintacticas_para_relaciones, especificaciones_sintacticas_para_entidades}

Vt= {APODA, DONDE, UNION, INTER, MENOS, VECES, REUNION,

DIVIDENTRE, [,], ::, :=, (,), {, }, /, ', NOT, T, F, <, >, =, <=>, <=, >=, A, ..., Z, a, ... , z, 0, ..., 9, ", +, -, ., DEFINE-TABLA CHARACTER, NUMERIC, LOGICAL, DATE, PROGRAM, BEGIN, END, : FOREIGN_KEY, REFERENCES, UPDATE OF, ON, ELSE, RESTRICTED, CASCADES, NULLFILES, PRIMARY_KEY, INSERT, UPDATE, REJET, FORSOME, DELETE, CHECK, CREATE_INTEGRITY_RULE, FORSOME, FORALL, IF, FOR, DROP_INTEGRITY_RULE, NULLS-[ALLOWED]-DELETE-OF}
S = {mini_relacional}

P= {<mini_relacional> → PROGRAM <id>;

<definiciones> BEGIN <bloque> END.

<definiciones> → <define_tablas><define_reglas>

<define_tablas> → <define_tabla> | <define_tabla><define_tablas>

<define_reglas> → <define_reglas> | <define_regla><define_reglas>

<define_regla> → <regla_primaria> | <regla_ajena> | ><crea_regla_a>

<regla_ajena> → FOREIGN KEY(<id>) | REFERENCES<objetivo>
NULLS-[NOT]-ALLOWED-DELETE-OF <objetivo><efecto>UPDATE-OF <claveprim_obj><efecto>

<objetivo> → <id>

<efecto> → RESTRICTED|CASCADES|NULLFILES |
<regla_integridad_particular>

<claveprim_obj> → <id>

<bloque> → <prog_alg>|<prop_alg><bloque>

<define_tabla> → DEFINE TABLA<alias>{<defcampos>;
<def_claveprim> | <def_claveajena>

<def_campos> → <def_campo> | [NOT]NULL<def_campo> |
<def_campo><def_campos>

<def_campos> → CHARACTER<nombre_atr>(<tamaño>); |
NUMERIC <nombre_atr>(<tamaño>, <decim>); |
LOGICAL <nombre_atr>; |
DATE <nombre_atr>;

<tamaño> → <numl>

<decim> → <numl>

<prop_alg> → <def_alias> | <asignación> | <regla_integridad>

<def_alias> → <alias> APODA <nombre_rel>;

<alias> → <id>
 <nombre_rel> → <id>
 <asignación> → <nombre_rel> [<lista_espec_atr>] := <exp_alg>;
 <lista_espec_atr> → <espec_atr> | <espec_atr>, <lista_espec_atr>
 <espec_atr> → <nombre_atr> | <nombre_rel> <nombre_atr> |
 <alias> . <nombre_atr>
 <nombre_atr> → <id>
 <def_claveprim> → PRIMARY KEY <id>; <regla_primaria> |
 PRIMARY KEY <id>;
 <def_claveajena> → FOREIGN KEY (<id>); | <def_reglaajena>
 <def_reglaajena> → REFERENCES <objetivo>
 NULLS-[NOT]-ALLOWED-DELETE-OF <objetivo>
 <exp_alg> → <selección> | <proyección> | <exp_infija>
 <campo_llave> → <id>;
 <selección> → <primitiva> DONDE <exp_bool>
 <primitiva> → <nombre_rel> | <alias> | (<exp_alg>)
 <proyección> → <primitiva> | <primitiva> [<lista_espec_atr>]
 <exp_infija> → <proyección> <op_infijo> <proyección>
 <op_infinito> → UNION | INTER | MENOS | VECES |
 REUNION | DIVIDENTE
 <exp_bool> → <oper_bool> | <opdo_bool> | NOT <opdo_bool>
 <oper_bool> → <opdo_bool> <opdr_bool> <opdo_bool>
 <opdo_bool> → (<exp_bool>) | (<comparación>) | <cte_bool>
 <comparación> → <espec_atr> <op_comp> <espec_valor>
 <espec_valor> → <espec_atr> | <cte>
 <cte_bool> → T | F
 <op_comp> → < > | = | < > | <= | >=
 <id> → <letra> | <letra> <cadena>
 <cadena> → <alfanúm.> | <alanum> <cadena>
 <alfanum> → <letra> | <dígito>
 <letra> → A | ... | Z | a | ... | z
 <dígito> → 0 | ... | 9
 <cte> → <cte_alfanum> | <cte_num> | <cte_bool>
 | <cte_date>
 <cte_alfanum> → "<cadena>"
 <cte_num> → <número> | + <número> | - <número>
 <número> → <numl> | <numl> . <numl>
 <numl> → <dígito> | <dígito> <numl>
 <cte_date> → '<numl>/<numl>/<numl>'
 <regla_integridad> → <crea_regla> | <borra_regla>
 <accion> → <accionl> | <accionl>, <accion>
 <accionl> → INSERT <situacion> | UPDATE <situacion>
 | DELETE <situacion>
 <cond_ok> → CHECK <ambito> <expr_booll>
 | CHECK <exp_booll>

```

<cond_no_ok> → <expr_bool1>
<expr_bool1> → <expr_bool2> | <expr_bool2><expr_bool1>
<expr_bool2> → <expr_bool> | <instruccion>
<crea_regla> → <nombre_regla><cond_ok><cond_no_ok>
               | <nombre_regla>ON <accion>;<cond_ok><cond_no_ok>
<nombre_regla> → CREATE_INTEGRITY_RULE <id_r>
<situacion> → <alias>.<nombre_atr>
<ambito> → FORALL <situacion> | FORSOME <situacion> | ...
<instrucción> → IF | FOR | ... | <instr_sql>...
<crea_regla_a> → <crea_regla> | <crea_r_rel>
<borra_regla> → DROP_INTEGRITY_RULE <id_r>
<id_r> → <id>
<regla_integridad_particular> → <disparador>

<regla_primaria> → <regla_relacional> | <regla_entidades> |
                  <crea_r_rel>
<crea_regla_a> → <crea_regla> | <crea_r_rel> ...
<crea_r_rel> → <regla_relacional_particular>
<regla_relacional> → <especificaciones_sintacticas_para_relaciones>
<regla_entidades> → <especificaciones_sintacticas_para_entidades>
    
```

Este último párrafo gramatical indica adentrarse en la investigación sobre definición sintáctica para relaciones y entidades. Consecuentemente, es posible extender la gramática para definir otro tipo de llaves y el tratamiento requerido de acuerdo a las necesidades de las aplicaciones de los usuarios.

7 Conclusiones

La extensión de la gramática propuesta permite observar cómo la teoría de Lenguajes Formales se puede enlazar a la Teoría del Modelo Relacional para desarrollar algunos elementos que puedan contribuir a solucionar uno de los problemas típicos en Bases de Datos: la solución a los problemas de Integridad.

En esta etapa del trabajo, las pruebas realizadas al intérprete de álgebra relacional con la incorporación de las extensiones gramaticales que permiten la definición y el tratamiento de restricciones de integridad, han sido satisfactorias.

Se espera que las especificaciones gramaticales desarrolladas en el presente trabajo, apoyen la construcción de componentes de programación que puedan incorporarse fácilmente dentro de los DBMS que requieran y permitan la incorporación de un tratamiento complementario para restricciones de integridad basadas en las especificaciones adicionales al modelo relacional presentadas en este trabajo. Por otra parte, también se espera que proporcionen un buen fundamento para el tratamiento de restricciones de integridad a nivel de las aplicaciones, posiblemente mediante procedimientos activados. Finalmente podemos citar que se tiene en proceso un conjunto de trabajos relacionados con el desarrollo de un lenguaje de integridad extendido y el desarrollo de un componente de integridad complementario para MySQL, entre otros.

Bibliografía

1. Codd. E.F. "A Relational Model of Data for Large Shared Data Banks". CAMC 13, núm.6 (junio 1970). Publicado otra vez Milestones of Reseca -Selected Papers 1958-1982 (CAMC, ejemplar del 25 aniversario), CACM 26, núm. 1 (enero de 1983).
2. Date C.J. "Introducción a los Sistemas de Bases de Datos". Vol. 1, 7ª. Ed. 2001, Pearson-Educación de México, S.A. de C.V.
3. Date C.J. "Introducción a los Sistemas de Bases de Datos". Vol. 1, 5ª. Ed. 1998, Addison Wesley Longman de México S.A. de C.V.
4. Ullman J.D. & J. Widom, "Introducción a los Sistemas de Bases de Datos". Prentice-Hall, México, 1999.
5. Galindo-Aburto Ercilia. "Integridad en DBMS Relacionales. Casos de Estudio: Access, MySQL, SQL Server.". Asesor: Ma. del Rocío Boone Rojas. Tesis de Licenciatura, Julio-2004, BUAP, Fac. de Cs. de la Computación, Pue. México.
6. Boone-Rojas Ma. del Rocío. "Especificaciones, Extensiones y Ramificaciones del Modelo Relacional", Proyecto de Investigación Interno. 2004, BUAP, Fac. de Cs. de la Computación, Secretaría de Investigación y Estudios de Posgrado. Pue. México.
7. Bernabé-Loranca Beatriz. "Procesador para Álgebra Relacional". Asesor: Ma. del Rocío Boone Rojas. Tesis de Licenciatura 1994, BUAP, Fac. de Cs. de la Computación, Pue. México.
8. Bernabé-Loranca, M.B. & Boone-Rojas, M. R.; "Especificaciones Complementarias para el Soporte de Integridad en DBMS relacionales", CIC 2004, Congreso Internacional de Computación, IPN, México, DF, Octubre 2004.
9. Boone-Rojas Rocío, Soriano-Ulloa M.A., Bernabé-Loranca, "Marco de Evaluación para el Componente de Integridad en DBMS relacionales". II Congreso Nal. de Cs. de la Computación. Puebla, Pue. Méx. Nov. 2004.
10. Pastor-López O, Pedro Blesa P.; "Gestión de Bases de Datos", Universidad Politécnica de Valencia. Servicios de Publicaciones, Valencia 2000.
11. Pastor-López, "Análisis de Restricciones de Integridad en el Nivel Conceptual", VII Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software (IDEAS 2004), Arequipa, Perú mayo 2004.

Manejo de información cartográfica mediante base de datos espaciales distribuida

G. Guzmán, M. Torres, R. Quintero, M. Moreno

Centro de Investigación en Computación - IPN, Av. Juan de Dios Bátiz,
esquina con Miguel Othón de Mendizábal, México, D.F., 07738. México
{guzmanl, mtorres, quintero, marcomoreno}@cic.ipn.mx

Resumen. El uso de Sistemas de Información Geográfica (SIG) ha crecido notoriamente debido a su importancia y relevancia en el manejo de información cartográfica. Actualmente hay SIG que permiten visualizar terrenos hasta sistemas más complejos en los cuales es posible identificar zonas de riesgo, realizar simulaciones de desastres naturales, etc. La mayor parte de dicha información, es almacenada en una base de datos para su empleo posterior. Existen aplicaciones comerciales, con las cuales es posible construir programas empleando algún lenguaje orientado a objetos, como lo es Java. Si bien es cierto, mediante Java se puede establecer una conexión con la mayoría de los manejadores de bases de datos actuales, la obtención de información tiene que hacerse de una base de datos en particular. En el presente trabajo, presentamos una arquitectura, que permite generar vistas geográficas integradas de datos descriptivos en los objetos geográficos, los cuales pueden estar almacenados bajo diferentes DBMS, dando la apariencia de una base de datos única, requiriendo como única condición, tener el mismo esquema físico en cada una de las bases de datos.

Palabras clave: Sistemas de Información Geográfica, Bases de datos distribuidas, información cartográfica, datos vectoriales, vistas geográficas.

1 Introducción

Hoy en día, la mayoría de las empresas privadas así como dependencias gubernamentales, se ven en la necesidad de consultar datos espaciales (datos asociados con la representación de un territorio en la Tierra) para realizar un determinado proyecto o una tarea en particular, por ejemplo, una empresa que desea construir una carretera de una ciudad a otra, requiere previamente información con gran exactitud referente al tipo de suelo, poblados aledaños, estructura topográfica, vías de comunicación, etc. Esto con el fin de que el proyecto a realizar no interfiera con alguna vía de comunicación o asentamiento ya existente.

Para el caso de cuando ocurre un siniestro provocado por un fenómeno meteorológico como puede ser un tornado o una inundación, las dependencias de protección civil, necesitan conocer los poblados que están seriamente afectados o pueden estar en peligro, para proceder a realizar una evacuación y con ello evitar la

pérdida de vidas humanas. Así como los dos ejemplos anteriores podríamos citar muchos casos más, en los cuales, los datos espaciales son de gran utilidad.

Desafortunadamente en México, muchas organizaciones realizan la obtención de datos espaciales manualmente, empleando mapas cartográficos en formato analógico, lo cual ocasiona entre otras cosas:

- a) Demora en los resultados. La mayoría de los mapas contienen demasiada información (curvas de nivel, vías de transporte, hidrología, poblados, etc.) por lo cual, el encontrar los datos requeridos se vuelve una tarea laboriosa y posiblemente la información se requiera rápidamente.
- b) Datos no actualizados. El mapa que se esté consultando puede tener 3, 5, 10 o más años de antigüedad, con lo cual, los datos obtenidos pueden ser erróneos o simplemente no encontrarse.
- c) Error en los cálculos. Se puede requerir de medir la distancia que hay entre dos objetos de interés, dado que los mapas generalmente están a escala, la medición manual seguramente dará un resultado demasiado impreciso.

Por lo anterior, diversas instituciones se han dado a la tarea de realizar la conversión de dichos datos en formato analógico a un formato digital que pueda ser almacenado en una computadora. Este proceso conocido comúnmente como digitalización permite consultar la información cartográfica desde una computadora, mediante aplicaciones específicas.

La problemática mayor se presenta cuando se tiene geo-información en formato digital, ya que esta información se encuentra dispersa y la información que se llegan a almacenar en una base de datos está en diferentes tipos de manejadores. Por tal motivo, el área de la Geocomputación actualmente se está enfocando en solucionar problemas de heterogeneidad e interoperabilidad en diferentes bases de datos geográficas.

En la presente propuesta se presenta una técnica que permite dar solución parcial al problema, mediante una arquitectura que permite obtener información almacenada en diferentes bases de datos. El resto del documento se encuentra organizado de la siguiente forma: en la sección 2 se describe la arquitectura propuesta para obtener e integrar información que se encuentra almacenada en diferentes bases de datos, la sección 3 describe el diseño de la arquitectura, la sección 4 muestra los resultados obtenidos con nuestra técnica propuesta y finalmente en la sección 5 se citan las conclusiones y trabajo futuro.

2 Arquitectura propuesta

La información almacenada en una base de datos, puede estar físicamente en un solo dispositivo, o bien, puede estar distribuida de acuerdo a alguno de los siguientes criterios [1]:

- a) BD distribuida horizontalmente. Son aquellas compuestas de fragmentos horizontales. Formalmente, un fragmento horizontal de una relación, es un subconjunto de las tuplas de esa relación. Las tuplas que pertenecen al fragmento

horizontal se especifican mediante una condición sobre uno o más de los atributos de la relación. Con frecuencia, solo interviene un atributo.

- b) BD distribuida verticalmente. Compuesta por fragmentos verticales de información, donde un fragmento vertical de una relación mantiene sólo ciertos atributos de la relación. Los fragmentos son ubicados en diferentes lugares.
- c) BD distribuida mixta. Es una base de datos que está distribuida tanto horizontal como verticalmente.

El lenguaje Java permite realizar conexiones a base de datos por medio del JDBC, pero comúnmente la información debe ubicarse en una sola base de datos. Desafortunadamente, no siempre todos los datos vectoriales que requiere una aplicación SIG se encuentran en una sola base de datos, más aún, las bases de datos pueden ser administradas por diferentes manejadores. A continuación, explicamos una arquitectura, conocida como GIS-DBC, la cual permite distribuir información vectorial de forma horizontal, la cual puede almacenarse en diferentes equipos y estar en distintos manejadores. En la Figura 1 puede observarse el esquema general de la arquitectura propuesta.

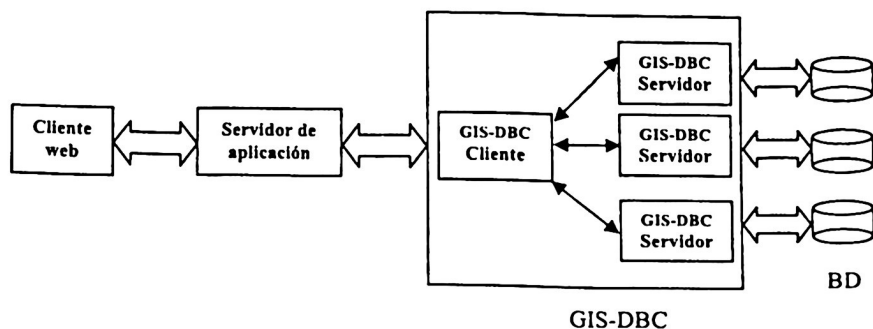


Figura 1. Esquema general de la arquitectura propuesta.

2.1 Cliente web

Utilizando algún navegador de Internet, el usuario accede a la aplicación para realizar alguna tarea o manipulación de datos vectoriales. Una vez identificada la información que se requiere, el cliente hace una petición al servidor de aplicación.

2.2 Servidor de aplicación

Su función principal es recibir la petición de información por parte del usuario final del sistema. Para proporcionar los datos requeridos, necesita obtener información de la base de datos vectorial, la cual solicita al GIS-DBC.

2.3 GIS-DBC

El GIS-DBC es el encargado de llevar el control de la base de datos distribuida horizontalmente. Al momento en que el servidor de aplicación realiza una consulta SQL, el GIS-DBC hace una búsqueda en cada una de las bases de datos, para poder encontrar la información de interés. Las dos partes más importantes de éste modulo son el GIS-DBC servidor y el GIS-DBC cliente, los cuales, se comunican empleando el mecanismo de RMI.

3 Diseño del sistema

En esta sección se describe el funcionamiento interno del GIS-DBC. Los componentes principales fueron ya previamente mencionados y son los GIS-DBC cliente y servidor.

3.1 El JDBC

Para poder realizar una conexión a una base de datos y ejecutar alguna consulta SQL ya sea para insertar, eliminar, modificar u obtener información almacenada, en un programa en Java, el JDBC define una serie de interfaces. Dichas interfaces, tienen que ser implementadas para cada manejador de base de datos con el que se desee establecer una comunicación. Actualmente, se pueden construir aplicaciones para acceder información a manejadores tales como: Oracle, Informix, SQL Server, MySQL, entre otros [2] [3]. Comúnmente, el proceso involucra la instanciación de objetos de las siguientes clases: *Connection*, *Statement* y *ResultSet*. El proceso de comunicación entre estos objetos, es tal como se muestra en la Figura 2.

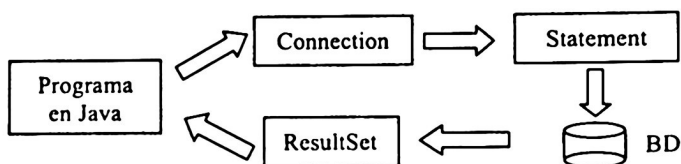


Figura 2. Mecanismo para realizar una consulta SQL mediante Java.

3.2 GIS-DBC cliente

El GIS-DBC cliente es un programa, que se ejecuta como proceso demonio, con la finalidad de revisar periódicamente, si existen peticiones de conexión a una base de datos. Cuando se desea ejecutar una instrucción SQL, por ejemplo de consulta, el GIS-DBC Cliente crea una instancia de la clase *GISDBCCConnection*. De manera breve, esta clase implementa la interfaz *Connection* y recibe como argumento una lista que contiene las direcciones IP de las máquinas en las cuales se encuentran las

bases de datos. En el proceso de inicialización del objeto, empleando RMI [4], se trata de establecer una conexión con el programa GIS-DBC Servidor de cada una de las máquinas.

El objeto *GISDBCConnection*, es el que se devuelve al programa que solicita la conexión a la base de datos. Posteriormente, se crean los objetos explicados en el punto anterior (de las clases *Statement* y *ResultSet*) para finalmente ejecutar la instrucción SQL deseada.

Lo importante de señalar es, no existe diferencia significativa, entre un programa que haga uso del GISDBC y de algún otro que realice conexión a una base de datos empleando JDBC, la única modificación radica en solicitar una conexión utilizando la clase *GISDBCDriverManager*, en lugar de la forma tradicional especificada en el lenguaje Java.

Lo anterior es la principal aportación de la arquitectura propuesta, con esto, las aplicaciones SIG previamente escritas, pueden migrarse rápidamente y sin invertir tiempo significativo para poder hacer uso de datos vectoriales que se encuentren en diversas bases de datos, claro está, cumpliendo la restricción de que tengan el mismo esquema físico y de preferencia no tener los mismos datos duplicados en distintas bases de datos.

Con respecto al punto anterior, inicialmente suponemos que los datos no se encuentran duplicados, de hecho, formalmente no sería una base de datos distribuida horizontalmente.

Cuando los datos se encuentran en distintas bases de datos, cada uno de los GIS-DBC Servidores devuelve la información vectorial que encuentra en la base de datos con la que está asociada, para finalmente, el GIS-DBC Cliente construir un objeto *GISDBCResultSet* que contiene todos los datos vectoriales solicitados.

3.3 GIS-DBC servidor

Es el encargado de comunicarse directamente con el manejador de base de datos. Un punto relevante y en el que se debe tener precaución, es que el GIS-DBC Servidor siempre debe estar en ejecución y cargarse antes de que el GIS-DBC Cliente quiera crear una instancia de la clase *GISDBCConnection* por primera vez, de lo contrario, no se podrá conectar a la base de datos en donde no se esté ejecutando en dicho instante, el GISDBC Servidor.

Supongamos que el GIS-DBC Cliente desea crear una instancia de la clase *GISDBCStatement* (la cual implementa la interfaz *Statement*), como no es posible conocer con anticipación de donde se obtendrá la información vectorial, se crea un *GISDBCStatement* que almacena en uno de sus atributos, una lista de las direcciones IP donde se encuentran las bases de datos. Esto con la finalidad de que cuando sea necesario ejecutar una instrucción SQL, dicha instrucción se ejecute en todos los servidores.

4 Pruebas del sistema

Actualmente, se tienen implementados adecuadamente, la mayoría de los métodos de las interfaces *Connection*, *Statement* y *ResultSet*. En la tabla 1 se muestran algunos tiempos de respuesta de dos programas distintos, en el primero de ellos, se empleó conexión directa empleando JDBC a una base de datos única en SQL Server, y en el segundo, usando la arquitectura desarrollada, teniendo dos bases de datos haciendo uso de los manejadores Informix y Oracle.

Tabla 1. Tabla comparativa de tiempos de respuesta.

Instrucción SQL	No. de Registros	Tiempo JDBC	Tiempo GIS-DBC
SELECT	10	2.25 ms.	3.84 ms.
INSERT	5	2.68 ms.	4.11 ms.
DELETE	4	2.30 ms.	3.91 ms.
UPDATE	5	2.54 ms.	5.01 ms.

Como puede apreciarse, el tiempo de ejecución haciendo uso de la arquitectura GIS-DBC es muy cercano a los resultados obtenidos mediante JDBC.

En la Figura 3 se muestra una aplicación escrita en Java, la cual obtiene información de un mapa digital de elevación (DEM por sus siglas en inglés) de diferentes bases de datos.



Figura 3. Visor de modelos digitales de elevación.

Un modelo digital de elevación consiste de un arreglo de muestras de altura medidas a partir del nivel del suelo, las cuales normalmente se encuentran espaciadas en intervalos constantes. Básicamente, un archivo DEM está integrado por tres tipos de registro, usualmente llamados A, B y C. La estructura de estos registros es la siguiente [5] [6]:

Registro A. Contiene información que define las características generales del DEM, incluyendo información de encabezado descriptiva, relacionada con el nombre del archivo DEM, bordes, unidades de medición, valores máximo y mínimo, el número de registros B y parámetros de proyección. Existe un solo registro A en el archivo DEM y aparece siempre como el primer registro.

Registro B. Contiene los datos de elevación y su información de encabezado relacionada. Todos los registros B de los archivos DEM son construidos con datos de bandas unidimensionales, llamados puntos. En consecuencia, el número de puntos totales que abarcan el área descrita por el DEM, es igual al número total de registros B que contiene el archivo.

Registro C. Contiene estadísticas de los datos en el archivo.

Asimismo se está trabajando en el desarrollo de una aplicación para la manipulación de datos espaciales. En esta aplicación se tienen implementados métodos visuales y espaciales como son: acercamiento (*zoom-in*), alejamiento (*zoom-out*), desplazamiento (*panning*), medición de distancia e identificación de atributos. En la Figura 4 se muestra una vista de la aplicación desarrollada.

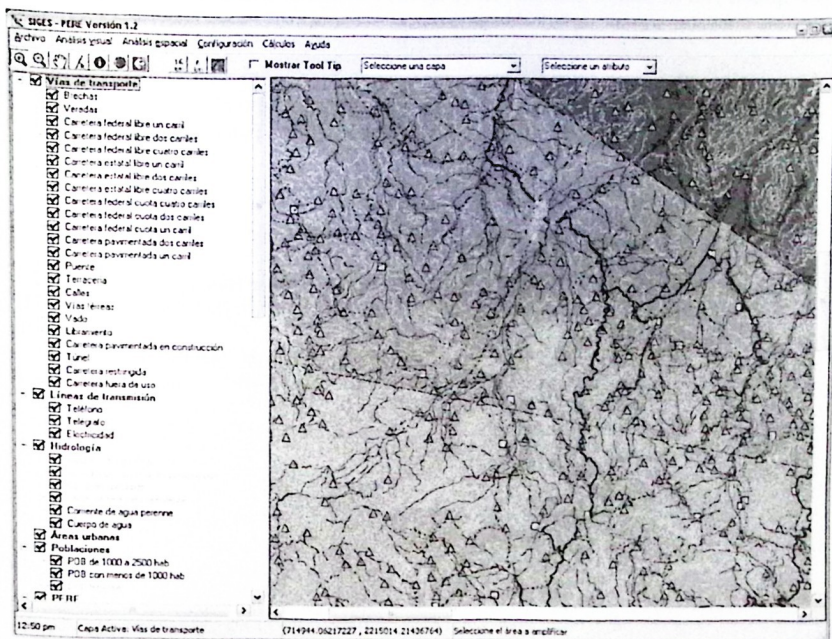


Figura 4. Herramienta SIG para manipulación de datos espaciales.

Esta aplicación fue desarrollada usando el Entorno Integrado de Desarrollo C++ Builder versión 5.0. Aprovechando la arquitectura desarrollada, la aplicación solicita la ejecución de un programa en Java para la obtención de información espacial almacenada en diferentes bases de datos, recopilada dicha información se procede a

guardarla en una base de datos única que es accedida por la aplicación principal para visualizar los datos completos.

5 Trabajo futuro

Existen algunos puntos que todavía requieren de trabajo de investigación para poder encontrar una solución completa. El primer punto, en el que nos encontramos trabajando actualmente, es el problema que surge cuando se necesita combinar información almacenada en diferentes bases de datos que debe combinarse para realizar operaciones más complejas como es la de producto cartesiano.

Otro punto de estudio, es cuando un servidor no se encuentra disponible por situación ajena al GIS-DBC Cliente, como puede ser, problemas en la red, el GIS-DBC Servidor no está en ejecución, entre otros. Como solución inicial, cuando no se puede establecer una conexión entre los GIS-DBC Cliente y Servidor, se lanza una excepción con la cual el usuario recibe un mensaje en el que se le indica que no fue posible conectar con un servidor en particular.

Agradecimientos. Los autores desean expresar su agradecimiento a los revisores del presente documento por sus comentarios pertinentes, así como al CONACYT, CGPI y al CIC – IPN por su apoyo económico para el desarrollo de esta investigación.

Referencias

1. Elmasri R & Navathe S, "Sistemas de bases de datos", Prentice Hall, pp. 704–713, 1999.
2. Deitel & Deitel, "Java How To Program Fifth Edition", Prentice Hall, ISBN: 0131016210, pp. 280-295, 2003.
3. William Grosso, "Java RMI", O'Really, ISBN: 1565924525, pp. 143-178, 2001.
4. Sun Microsystems, "Tutorial de Java", <http://www.java.sun.com>.
5. U.S. Geological Survey, <http://edc.usgs.gov/>.
6. <http://www.gisdatadepot.com/dem/>
7. <http://www.mapmart.com/DEM/DEM.htm>.

Hardware and Control

Hardware y control

Diseño y construcción de un sistema telemétrico de registro extracelular multiunitario

Francisco J. Heredia López¹, Br. Zoemy Esquivel Loria²

¹Universidad Autónoma de Yucatán, Centro de Investigaciones Regionales "Dr. Hideyo Noguchi". Laboratorio de Ingeniería Biomédica. hlopez@tunku.uady.mx

²Universidad Autónoma de Yucatán, Facultad de Matemáticas. zgel_09@yahoo.com.mx

Resumen. Este trabajo describe los avances que se han logrado en el proyecto de desarrollo de un sistema para la transmisión inalámbrica de potenciales de acción neuronales extracelulares a una PC, y que permita mover mediante señales inalámbricas un conjunto de hasta cuatro electrodos implantados en el cerebro de la rata, sin tener que inmovilizar o anestesiarse al animal.

Abstract. This paper describes the advances that have been achieved in the project of development of a system for the wireless transmission of extracellular action potentials to a PC, and that allows to move a group of up to four electrodes with wireless commands without having to immobilize or to anesthetize the animal.

1 Introducción

Uno de los problemas que con más frecuencia obstaculizan la medición de la actividad eléctrica de las neuronas es la vibración a la que están sometidos los microelectrodos durante el movimiento de los animales, lo cual hace que la punta del electrodo cambie de posición y se pierda la señal. Esto se ha tratado de solucionar mediante dispositivos de avance micrométrico que permiten reubicar los electrodos para localizar otra neurona o estructura funcional, pero que poseen la desventaja de ser controlados manualmente, lo cual requiere manipular e inmovilizar al animal, alterando su comportamiento espontáneo, con lo que se reduce la validez y confiabilidad de los datos.

En el Laboratorio de Ingeniería Biomédica del Centro de Investigaciones Regionales de la UADY, se está desarrollando un sistema para la transmisión inalámbrica de potenciales de acción neuronales extracelulares, teniendo como objetivo el desarrollo de un sistema telemétrico implantable en el cráneo de ratas, que permita mover un conjunto de hasta cuatro electrodos sin tener que inmovilizar o anestesiarse al animal y que transmita en forma inalámbrica las señales de la actividad eléctrica a una computadora cercana.

Actualmente se han desarrollado los circuitos para establecer comunicación por Radio-Frecuencia (RF) utilizando la tecnología RF-Monolithics. La señal a transmitir es previamente digitalizada por un microcontrolador PIC12CE674, para posteriormente enviarla de manera serial utilizando el transceptor DR3000-1 de RF-Monolithics logrando tasas de transferencias de 125 Kb/s. La señal es recuperada

utilizando el microcontrolador PIC16F877, el cual establece comunicación con la PC a través del puerto paralelo; dicha señal es visualizada en la PC utilizando una interfase creada en el lenguaje LabView 6.1.

2 Antecedentes

En trabajos previos se han reportado diversos sistemas para registrar la actividad eléctrica extracelular mediante arreglos de microelectrodos implantados en el cráneo, los cuales son ubicados en el núcleo cerebral de interés por dispositivos de avance micrométrico controlados manualmente [1] [2].

En todos estos trabajos se resalta la importancia de poder medir varias unidades simultáneamente con el fin de correlacionar de manera más completa la actividad eléctrica detectada con las respuestas a los estímulos empleados durante el proceso experimental. Sin embargo, los dispositivos reportados poseen varios inconvenientes que limitan la posibilidad de realizar un mapeo de las distintas capas de neuronas presentes en la región cerebral en estudio, sin manipular y alterar el comportamiento de los animales, lo cual restaría validez a los resultados.

En algunos estudios [1] [3], los sistemas desarrollados se han basado en el uso de dispositivos capaces de detectar varias unidades de manera simultánea y de reubicar manualmente el arreglo de microelectrodos en caso de pérdida de la señal o de querer realizar registros a distintos niveles (capas de tejido), pero con el inconveniente de que la información es transmitida de manera alámbrica [4], lo cual impone ciertas restricciones al movimiento del animal en experimentación.

Es por ello que algunos autores han enfocado sus esfuerzos al diseño de sistemas para la captura y transmisión inalámbrica de las señales eléctricas [5] [6]. Cabe destacar que en los trabajos citados, el cambio de posición de los arreglos de microelectrodos sólo podía efectuarse manualmente, introduciendo involuntariamente en el animal alteraciones conductuales, ya sea por la propia manipulación o por la administración de sedantes o anestésicos que alteran la actividad eléctrica de grupos neuronales, tal como ha sido reportado en otros estudios [7] [8].

Como puede observarse, los enfoques de los trabajos reportados en la literatura no convergen en una metodología capaz de registrar la actividad eléctrica neuronal mediante microelectrodos de registro con avance micrométrico controlado a distancia y que permita capturar por vía telemétrica la señal electrofisiológica, sin tener que restringir la actividad del animal o introducir alteraciones en los patrones de actividad eléctrica por la administración de sedantes o anestésicos.

El proyecto en proceso tiene como objetivo desarrollar un implante en el cráneo del animal que contenga:

- (a) Un arreglo de hasta cuatro electrodos que permita detectar simultáneamente los potenciales de acción de varias neuronas.
- (b) Un sistema electromecánico en miniatura controlado de manera inalámbrica que permita reubicar el avance de los electrodos en pasos precisos y conocidos para localizar a otras neuronas.

Además incluirá un dispositivo electrónico en miniatura para la amplificación y transmisión inalámbrica de las señales electrofisiológicas a un receptor cercano, con el fin de tener al animal sin restricciones físicas que impidan su libre movimiento. Un segundo módulo receptor-transmisor que reciba las señales provenientes del módulo en el cráneo de la rata y que transmita la información necesaria para reposicionar la tétada de microelectrodos (montados en un portaelectrodos).

En ambos casos la información será enviada o provendrá de una PC en el caso de que el usuario desee mover la tétada de electrodos.

3 Métodos

Actualmente se han diseñado e implementado los circuitos electrónicos y los programas de computadora que constituyen un sistema automático para la transmisión inalámbrica de potenciales de acción extracelulares y para el control inalámbrico de reposicionamiento de los micro electrodos.

Así el trabajo desarrollado puede centrarse en dos grandes secciones: Desarrollo del "hardware" y desarrollo del "software".

3.1 Desarrollo del hardware

3.1.1 Módulo Transmisor

Actualmente éste módulo está construido alrededor de un microcontrolador RISC ("Reduced Instruction Set Computer") PIC12CE674, para digitalizar los potenciales de acción provenientes de los amplificadores. Para lograr un enlace telemétrico digital por señales de radiofrecuencia se emplea un transmisor de tecnología RF-Monolithics, específicamente el transceptor DR3000-1 en modo transmisor (ver figura 1).

Así, el convertidor análogo digital (A/D) de cuatro canales del microcontrolador hace un muestreo de alguno de los cuatro canales de los amplificadores y cada "byte" obtenido es transmitido de manera serial, logrando una tasa de transferencia de 125 Kb/s utilizando el protocolo RS-232, esto es, se logra digitalizar y transmitir 21 muestras por canal, teniendo por cada evento una duración de 2ms.

3.1.2 Circuito preamplificador

La señal eléctrica proveniente de la actividad neuronal, es enviada a este circuito preamplificador. El diseño de este circuito está basado en un seguidor de voltaje J-FET (con un transistor 2N5457) para lograr una impedancia de entrada de varios Mohms. La salida de este seguidor se acopla a una etapa de alta ganancia de voltaje construida por medio de un transistor "darlington" MPSA13. Esta etapa tiene una ganancia aproximada de 10, teniendo un consumo de energía de 833 μ A.

Para las pruebas mostradas en este trabajo, la señal que recibe el preamplificador proviene de un circuito generador de señales que simula ráfagas de potenciales de acción.

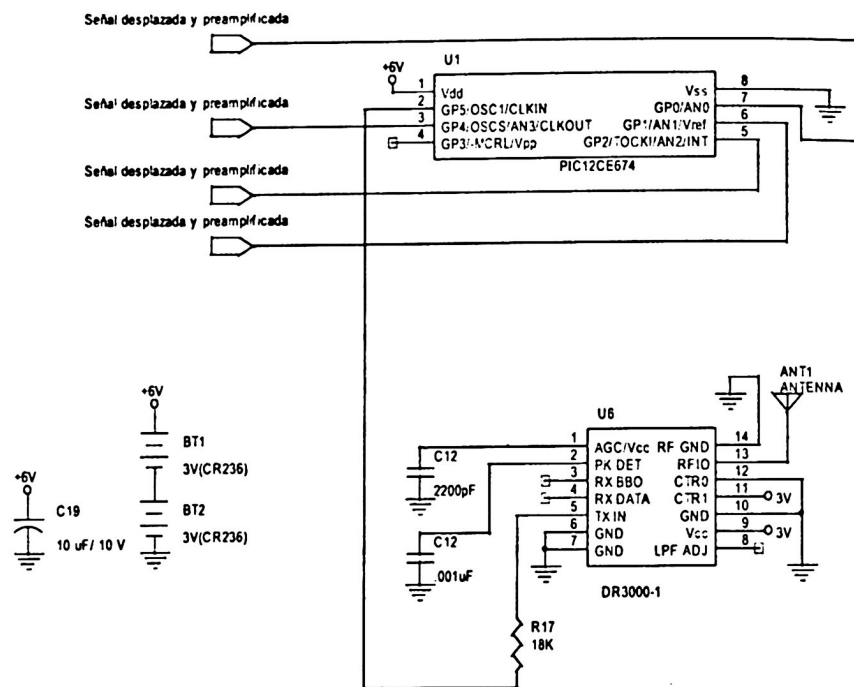


Figura 1. Diagrama del módulo transmisor.

3.1.3 Circuito amplificador y desplazador de nivel

La señal amplificada de los potenciales de acción es pasada a través de un desplazador de nivel con ganancia de 100 a fin de montarla sobre un voltaje constante de la mitad del valor de la fuente de alimentación, para de ahí enviarla al microcontrolador y digitalizarla, esto es debido a que el convertidor A/D del PIC12CE674 no maneja voltajes negativos.

3.1.4 Circuito generador de señal

Para probar las secciones previamente descritas (preamplificador, desplazador de nivel) se llevó a cabo el diseño y construcción de un circuito generador de pulsos basado en el microcontrolador Microchip PIC16C84, que es capaz de generar cuatro patrones de pulsos diferentes, cada patrón con una duración de 20ms, aproximadamente, los cuales son enviados a cuatro derivadores y de ahí a un divisor resistivo, obteniendo así pulsos con flancos de salida exponenciales negativos de 2ms de duración con una amplitud de 5mV aproximadamente, que simulan un potencial de acción.

La figura 2 muestra el diagrama de los circuitos desplazador, preamplificador de señal y generador de señal para un canal.

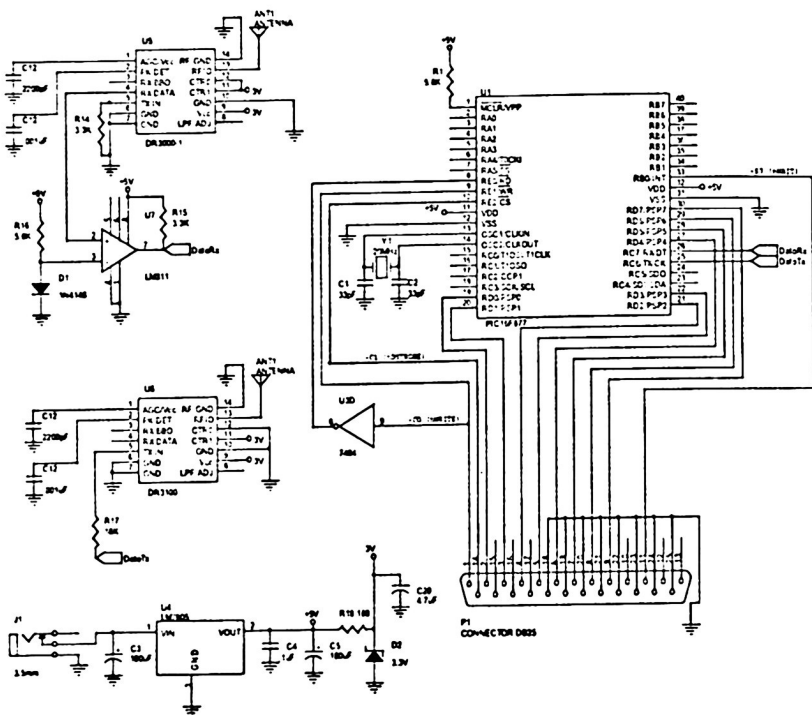


Figura 3. Diagrama del módulo receptor.

3.2 Desarrollo del software

3.2.1 Módulo Transmisor

El programa desarrollado para el microcontrolador que se utiliza en el módulo transmisor ejecuta las siguientes tareas:

1. Programa el convertidor A/D para utilizar los bits GP0, GP1, GP2 y GP4 (canales 0 al 3 del A/D) como entradas de la señal a digitalizar provenientes de la tetrada de electrodos.
2. Programa el bit GP5 como la terminal de salida de los datos digitalizados hacia el transmisor de radio frecuencia.
3. Programa el Puerto GP3 como terminal de entrada digital para recibir comandos de cambio de canal a digitalizar y transmitir.
4. Programa el convertidor A/D para que digitalice en los canales 0 a 3, con una frecuencia de conversión de 500KHz, con lo que se obtiene un tiempo de conversión (TAD) de $1/500 = 2$ microsegundos.

5. Programa el registro OPTION REGISTER para que el "timer" (TMR0) decremente su conteo con una frecuencia de 500 KHz (una cuenta cada 2 microsegundos).
6. Programa el TMR0 con 250 para que al llegar a 0 hayan transcurrido 12 μ s y se limpie la bandera de rebose del mismo (TOIF). Estos 12 μ s se requieren para que el A/D pueda muestrear correctamente el canal a digitalizar.
7. Seguidamente el bit GO del registro ADCON0 del A/D es puesto en alto para iniciar la digitalización del canal seleccionado, y el programa vigila en un lazo de espera cuando este bit va a bajo (digitalización completa, dato disponible en ADRES).
8. Al detectar el fin de conversión el programa ejecuta una rutina destinada a cambiar el valor binario en los bits ADCON0,3 y ADCON0,4 para seleccionar el siguiente canal (electrodo) y pasa el resultado de la digitalización a la variable l_dummy por medio del registro de trabajo W, e inicia otra digitalización (pone GO en H).
9. Seguidamente el programa transmite el dato en la variable l_dummy por GP5 usando el protocolo RS232 a 125Kb/s a 8n1s. La rutina está escrita en forma lineal. Al terminar de transmitir el dato, el programa chequea por si hay solicitud de cambiar de canal, y salta a la rutina de actualización de canal a digitalizar (paso 8) y el proceso se repite.

3.2.2 Módulo Receptor

El programa desarrollado para el microcontrolador que se utiliza en el módulo receptor ejecuta las siguientes tareas:

1. Programa los puertos A y C como entradas y los puertos B y D como salidas.
2. Programa el puerto paralelo esclavo del microcontrolador.
3. Programa el puerto serial del PIC16F877 para una recepción asíncrona a una tasa de transferencia de 125 Kb/s, sin paridad, un bit de stop, 8 bits de datos.
4. Inicia la recepción de datos seriales provenientes del receptor de radiofrecuencia.
5. Checa si hubo algún error durante la recepción (mediante recepción completa). Si es así, lo intenta de nuevo.
6. El microcontrolador chequea la bandera del buffer de recepción de su puerto paralelo para detectar la presencia de un byte de comando proveniente de la PC.
7. Si hay dato recibido el microcontrolador salta a una rutina para recibir otros dos bytes. En caso de que el primer byte recibido sea una M, los siguientes dos bytes contienen el total de micras que los microelectrodos avanzarán. Si por el contrario el primer byte recibido es una C, el siguiente byte contiene el número de canal a digitalizar.
8. Si la bandera de recepción del puerto paralelo esclavo está inactiva, el dato válido proveniente del puerto serial es transmitido por el microcontrolador a la PC por medio de su puerto paralelo esclavo usando el protocolo EPP.

3.2.3 Módulo generador de señal

Como se mencionó en la sección del transmisor se construyó un circuito generador de señales de potenciales de acción, usando un microcontrolador PIC16C84, el cual es controlado por un programa que realiza las siguientes funciones:

1. Inicializar el puerto B como salida, y el oscilador como RC a 4MHz.
2. Se programa el OPTION REGISTER para que trabaje con una frecuencia de 1:256
3. Se programa el TMR0 para que cuente 2 ms.
4. Si ya pasaron los 2 ms:
 - 4.1 Generar 4 patrones de pulsos diferentes, alternando 0's y 1's
5. Programar el TMR0 para contar de nuevo 2ms y se regresa al paso 3.

3.2.4 Programa para la recuperación de los datos

El programa para recuperar las señales en una PC fue desarrollado en lenguaje gráfico LabView 6.1 de National Instruments, el cual realiza las siguientes tareas (ver figuras 4,5 y 6):

1. Poner en bajo C0, C1 y C3 del puerto de control del puerto paralelo, antes de acceder al registro EPP e iniciar la recepción de datos.
2. Inicializar con ceros un arreglo de datos con capacidad de 1024 datos.
3. Se lee constantemente S7 del Puerto de Estatus hasta que el dato leído sea 128 (bit 7 en alto) lo cual significa que existe un nuevo dato en el Puerto Paralelo Esclavo del microcontrolador.
4. Leer el dato en el registro de datos (dirección base + 4).

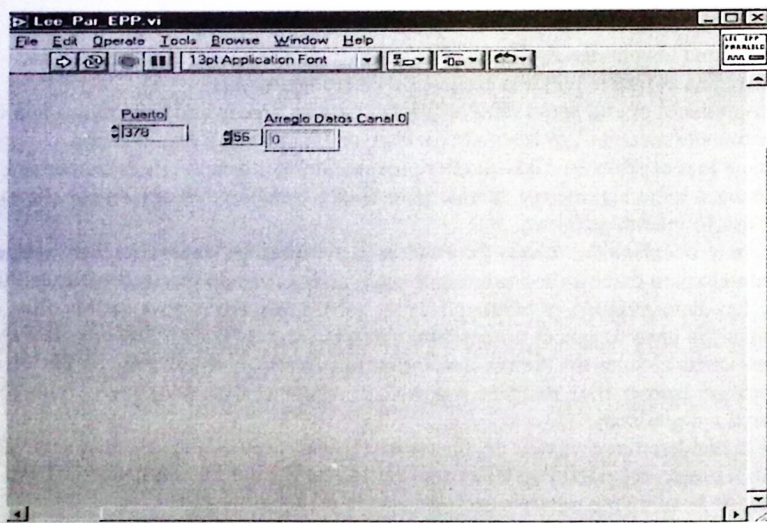


Figura 4. Inicialización del programa de recepción de datos.

5. Pasa el arreglo de datos a una función de graficación para su desplegado.

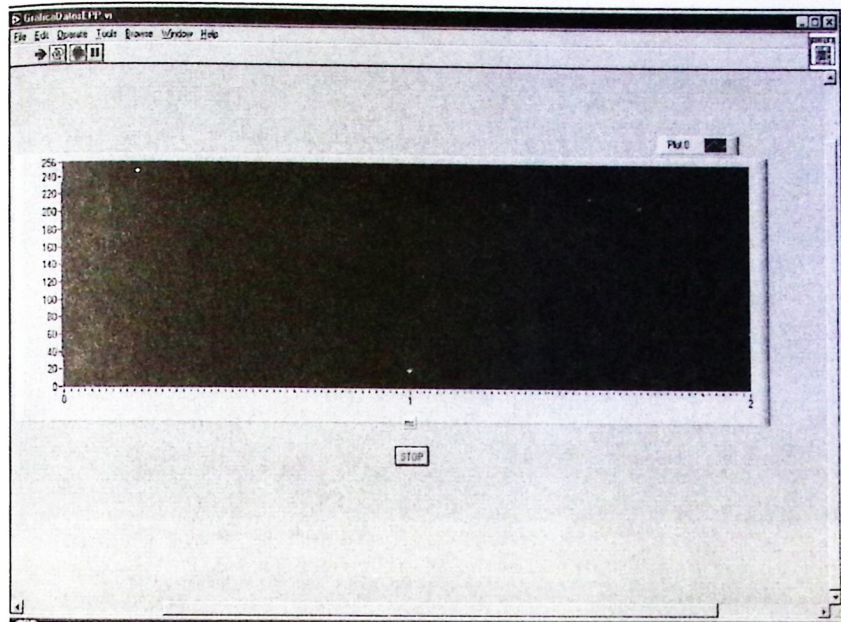


Figura 5. Pantalla para la representación gráfica de la recepción de datos.

4 Resultados

En las primeras pruebas se digitalizaron cuatro canales de manera secuencial utilizando una señal senoidal con una frecuencia de 500 Hz y obteniendo 8 muestras por evento. Esta cantidad de información es insuficiente para obtener una imagen digitalizada de un potencial de acción con una adecuada resolución. Por trabajos previos, en las condiciones experimentales se necesita un mínimo de 10 muestras por evento.

Para las pruebas actuales se utilizó únicamente el canal 0 del sistema desarrollado obteniendo 21 muestras por evento, situación que permite una digitalización de un potencial de acción con buena resolución. (Ver figura 6).

Posteriormente se sustituyó la señal senoidal, por la señal proveniente del generador de patrones. En la figura 7 se observa en el osciloscopio la señal proveniente del generador de patrones; y en la figura 8 se muestra a modo comparativo la señal recuperada dentro de la PC.

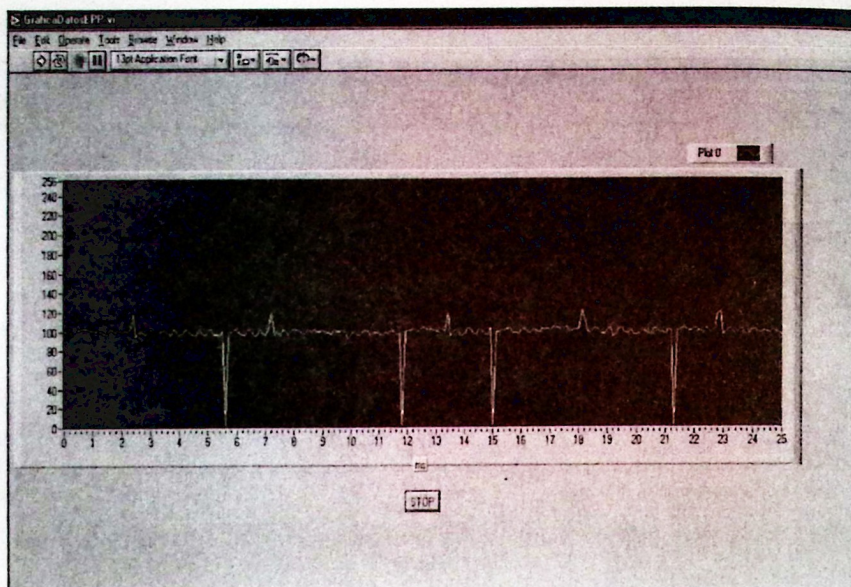


Figura 6. Señal senoidal recuperada en la PC.

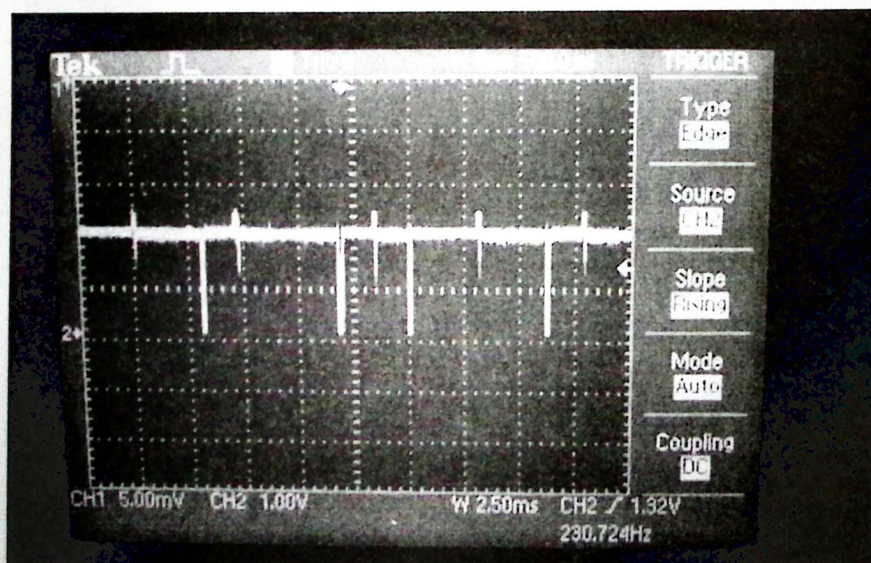


Figura 7. Señal proveniente del generador de pulsos.

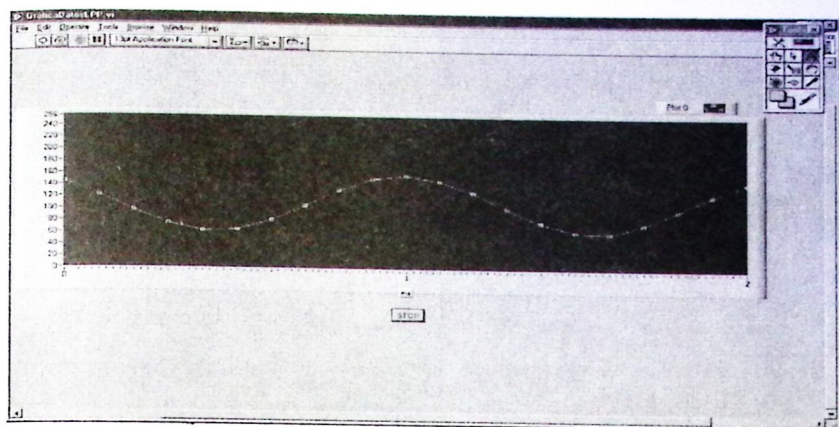


Figura 8. Señal recuperada en la PC

En la siguiente figura (figura 9) se muestra una imagen de la tarjeta del receptor e interfase con la PC, pueden observarse los módulos de transmisión y recepción por RF.

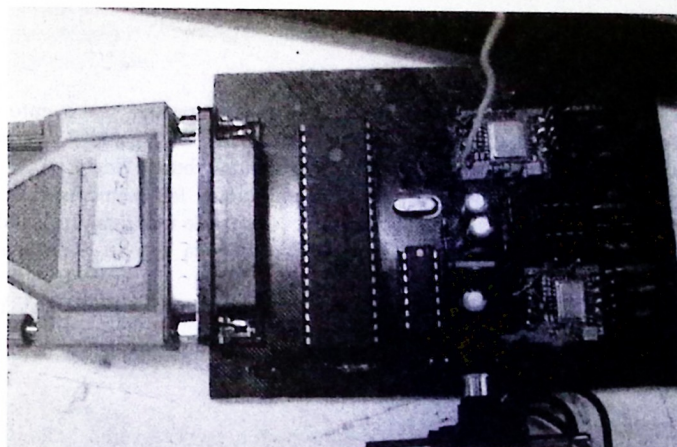


Figura 9. Módulo actual de la etapa receptora.

En la figura 10 se muestra una imagen de la tarjeta del sistema transmisor integrado, el cual incluye las etapas: preamplificadora, amplificadora y desplazadora de nivel, digitalizadora y transmisión por Radio Frecuencia a través del transceptor DR3000-1.

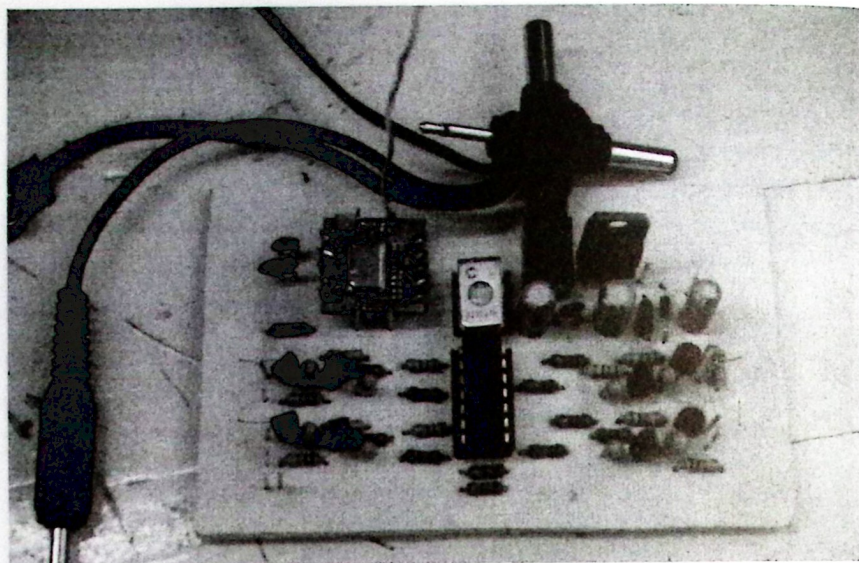


Figura 10. Módulo actual del sistema transmisor.

5 Discusiones

Como puede observarse en la figura 6 se obtienen 21 muestras por evento de 2 ms, este número de muestras al hacer la comparación entre las figuras 7 y 8 permite la reconstrucción del potencial de acción simulado con muy buena fidelidad.

La siguiente etapa del proyecto consiste en iniciar las primeras pruebas para obtener las señales eléctricas provenientes directamente de una rata, utilizando ambos sistemas: transmisor y receptor y visualizar en la PC la actividad eléctrica de las neuronas.

Este proyecto está patrocinado por el convenio Sectorial SEP/CONACYT 2003-2006 No.2003-01-31-001-056.

Bibliografía

1. Ainsworth A, O'Keefe JA. Lightweight microdrive for the simultaneous recording of several units in the awake, freely moving rat. *J Physiol (Lond)*, 269(1):8P-10P, 1977.
2. V.A. Korsunov. Miniature microdrive for extracellular recording of neuronal activity in freely moving animals.. *Journal of Neuroscience Methods*, 77-80, 1994.

3. Imre Szabó a, András Czurkó, Jozsef Csicsvari, Hajime Hirase, Xavier Leinekugel, Gyorgy Buzsaki. The application of printed circuit board technology for fabrication of multi-channel micro-drives. *Journal of Neuroscience Methods* 105,105–110, 2001.
4. Mountcastle VB, Reitboeck HJ, Poggio GF, Steinmetz MA. Adaptation of the Reitboeck method of multiple microelectrode recording to the neocortex of the waking monkey. *J Neurosci Meth* ,36:77-84, 1991.
5. Eichenbaum H, Pettijohn D, Deluca AM, Chorover SL. Compact miniature microelectrode-telemetry system. *Physiol Behav*,18(6):1175-8, 1997.
6. H. Fisher, H. Kautz, W. Kutsch. A radiotelemetric 2-channel unit for trnasmission of muscle potentials during free flight of the desert locust, *Schistocerca gregaria*. *Journal of Neuroscience Methods*, 39-45, 1995.
7. Mark D. Kelland, Louis A. Chiodo, Arthur S. Freeman. Dissociative Anesthesia and striatal neuronal electrophysiology. *Synapse* , 75-78, 1991.
8. Mark O. West. Anesthetics eliminate somatosensory-evoked discharges of neurons in the somatotopically organized sensorimotor striatum of the rat. *The Journal of Neuroscience.*; 9055-9068, 1998.
9. LabVIEW 6.1. User Manual. National Instruments Corporation. 1997
10. Microchip Microcontroller PIC16F877, 1-218, 2001.
11. Microchip Microcontroller PIC16CE674, 1-218, 2001.

Several Control Strategies for Parallel Connected Dual Induction Engines by means of a Digital-Based System Control

R. Peña-Eguiluz,^{1,2} M. Pietrzak-David,² B. de Fornel²

¹ Instituto Nacional de Investigaciones Nucleares, Laboratorio de Física de Plasmas
AP 18-1027, 11801 México D. F., México

² Laboratoire d'Electrotechnique et d'Electronique Industrielle
Unité Mixte de Recherche INPT-ENSEEIH / CNRS N° 5828
B.P. 7122 - 2 rue Camichel - 31071 Toulouse Cedex 7 - Toulouse, France
rpe@nuclear.inin.mx

Abstract. This paper deals with the description, implementation and simulation of four different digital control circuits proposed for a single voltage-source inverter (mono-inverter) supplying two parallel connected induction (dual induction) engine systems. Several losses in the degrees of freedom result from the connection of two motors in parallel. Therefore, four non-conventional DSP-FPGA based control strategies are proposed in this paper in order to compensate for the restrictions imposed by the high-power structure. A brief description of the proposed high-power system model, representing a commercial locomotive and its associated digital-control structure is presented. The principles of four control methodologies are discussed and analyzed. With an aim to demonstrate the resulting system operation, one mechanical perturbation has been applied to it, namely the loss of adherence of a wheel to its rail. The four system responses are assessed by simulation and compared among themselves, establishing the principal differences between the four proposed digital control methods.

1 Introduction

The multi-converter multi-machine systems (MMS) are used in several industrial applications. One of these systems may be implemented by a mono-inverter dual induction machine working with a common mechanical load [1, 2]. The structure goal consists in the quantity and size reduction of the required power electronics and control system components, decreasing the maintenance cost and by the way the price of construction [3]. This kind of structure can be applied in several industrial areas like electrical propulsion systems that have components on board, such as the robots, bus, ship, electrical and hybrid vehicles. It can be extended and applied on speed or positional control of usually conveyors.

In this paper a simulation model with a particular configuration is represented by a mid-bogie of a railway traction locomotive. It is powered by two similar asynchronous motors, which are supplied by a single voltage-source inverter (VSI). A DSP-FPGA based system is used as a digital unit control. A DSP processes the measured engines signals (currents and mechanical speeds) in order to capture the appropriated

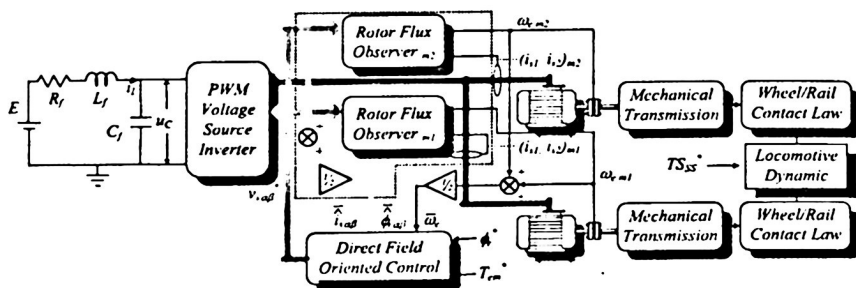


Fig. 1. The high power system working with a dual rotor flux observer structure.

semiconductor firing signal sequence from a modified Field Oriented Control (FOC) working with an innovative rotor flux full-order observer structure. Meanwhile, a FPGA produces the firing pulses to the semiconductors of the VSI at an appropriated frequency rate, which is always higher than the voltage frequency applied to the engines, provided that a Pulse Width Modulation (PWM) technique is applied to control the VSI. This kind of system is feasible if and only if an adequate control maintains an equal electromagnetic torque on the two engines even if their speeds differ.

2 System description

The model representation of the high-power system that has been studied is shown in Fig. 1. It is mainly composed by two similar squirrel cage induction engines, supplied by a PWM controlled VSI. A DC voltage source (E) is connected to the inverter through a passive (R_f , L_f and C_f) input filter. A digital control unit made from a DSP-FPGA association takes care of the rotor flux observation, the electrical rotor speed measurement, the control algorithm and the semiconductors commutation signal generation. Each engine drives an identical mechanical transmission line, which provides the traction force transmitted to the wheels. A wheel-rail non-linear contact law, defines this traction force [4]. The locomotive dynamics is common to both engines, so that they are linked mechanically. A singular characteristic is established in this kind of structure because both engines are electrically and mechanically linked by the VSI, reducing the control degrees of freedom, and by the mechanical load, respectively. The FOC high performance function is made possible when appropriated electromagnetic engine information is provided to it. Thus, in order to control properly the high-power system, several FOC algorithm adaptations have to be carried out.

3 Rotor Flux Estimation

Many industrial high-performance engine applications depend on the adequate knowledge of the internal engine state. Taking into account the excessive cost in-

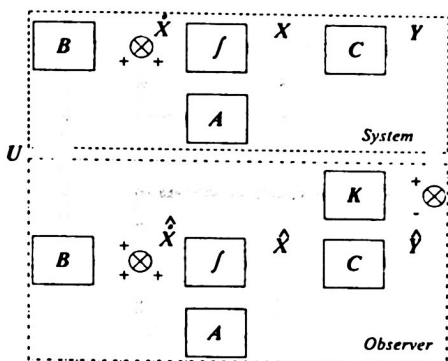


Fig. 2 General deterministic observer structure.

crease that normally occurs when a flux detector is incorporated to the engine, the use of an indirect flux detection, e.g. the rotor flux full-order deterministic Luenberger observer, is often desirable. Its general structure is shown in Fig. 2.

This kind of observer estimates the stator current and the rotor flux components simultaneously. It can be determined in a stationary reference frame by the next equation:

$$\dot{\hat{X}} = (A - KC)\hat{X} + BU + KCX \quad (1)$$

where:

$$X = \begin{bmatrix} i_{sa} \\ i_{sb} \\ \phi_{ra} \\ \phi_{rb} \end{bmatrix}, \quad A = \begin{bmatrix} a_1 & 0 & a_2 & a_3\omega_r \\ 0 & a_1 & -a_3\omega_r & a_2 \\ a_4 & 0 & a_5 & -\omega_r \\ 0 & a_4 & \omega_r & a_5 \end{bmatrix}, \quad K = \begin{bmatrix} k_1 & -k_2 \\ k_2 & k_1 \\ k_3 & -k_4 \\ k_4 & k_3 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \sigma L_s & 1 & 0 & 0 \\ 0 & \sigma L_s & 0 & 0 \end{bmatrix}^T, \quad U = \begin{bmatrix} v_{sa} \\ v_{sb} \end{bmatrix},$$

$$a_1 = \left(\frac{1}{\sigma\tau_s} + \frac{1-\sigma}{\sigma\tau_r} \right), \quad a_2 = \frac{1-\sigma}{\sigma\tau_r L_m}, \quad a_3 = \frac{1-\sigma}{\sigma L_m}, \quad a_4 = \frac{L_m}{\tau_r}, \quad a_5 = -\frac{1}{\tau_r}.$$

The observer's gain matrix, which is operated as $A-KC$, determines the observer convergence dynamics. Values of the gain matrix K have been calculated by applying a novel approach. It consists in establishing the observer response or 'observer root-locus' (ORL) to be proportional to the 'induction engine root-locus' (IMRL). This adaptable observer is defined by the calculated flux observer gain matrix values defined by

$$\begin{aligned}
k_1 &= d \\
k_2 &= (1 - k) \omega_e \\
k_3 &= \frac{b_1}{a_3} + \frac{a_2(b_2 - a_2 b_1)}{a_3 a_2^2 + a_3^3 \omega_e^2} \\
k_4 &= \frac{\omega_e(b_2 - a_2 b_1)}{a_2^2 + (a_3 \omega_e)^2}
\end{aligned} \tag{2}$$

where:

$$\begin{aligned}
b_1 &= d \left(\frac{k}{2} - 1 \right) + (1 - k)(a_1 + a_3 a_4 - a_5), \\
b_2 &= a_3 \left[\frac{d}{2} \left(\frac{d}{2} - a_1 + a_5 \right) + (k - 1) \omega_e^2 \right].
\end{aligned}$$

So, the observer dynamics mainly depends on the electric engine speed, on several engine parameters and on two adjusting coefficients. The displacement coefficient d can be adjusted by increasing or reducing the observer time response. This becomes faster than the system time response as d is smaller than the unity. It means that the *ORL* origin can be shifted, from the *IMRL* origin, $d/2$ real units towards the original complex plane or towards the negative complex plane, reducing the observer time response. The damping coefficient k can be altered in order to increase or reduce the real component of the dominant complex poles. The observer damping factor can be increased if k is smaller than the unity. As a result, the system performance will be improved reducing the oscillation effects of the engine response.

It can be seen from Fig. 3 that the *ORL* has a slower time response than the *IMRL*, because it is placed ten units toward the negative complex plane. Furthermore, the *ORL* has a better damping factor than the *IMRL*, because its complex values have lower angles with respect to the real axis. Both root-locus responses have been calculated as a function of the electrical rotation speed ω_F , which has been varied from 0 to 100 rad/s. Meanwhile, the *ORL* was obtained considering the same range of ω_F , and imposing the coefficients $k = 0.5$ and $d = 20$. In order to increase the time response of the implemented system, it is necessary to apply a d coefficient lower than the unity, so that it was chosen to be 0.9. Meanwhile, the other coefficient, k , was assumed equal to the unity. Consequentially, the response of the observer became accelerated in relation to the engine response, without any damping modification.

4 Dual Motor Drive Control Structures

The novel dual engine drive control configurations presented in this paper are based on the classical FOC method, although several FOC adaptations must be imposed on each one of these structures. As every specific system regulation produces several differences on the system behavior, the system under operation could be altered in different ways when a perturbation arrives. The four different studied control methods are briefly described as follows:

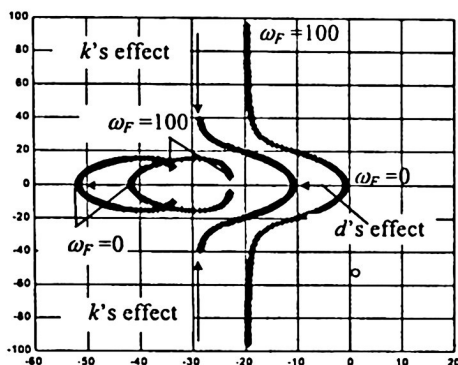


Fig. 3. Induction motor and observed root-locus as a function of the reference frame frequency.

4.1 Mean Drive Control

The mean drive control structure, shown in Fig. 1, takes into consideration the necessary signals of both engines in order to recreate an “imaginary mean engine” with the “average engine variables”. It has implemented using the “dual rotor flux observer” as it provides the most adequate flux estimate [5]. The observed variable space state vector and the electrical rotor speed used in this structure are defined respectively by:

$$\mathbf{X} = \begin{bmatrix} \frac{i_{sa_m1} + i_{sa_m2}}{2} & \frac{i_{s\beta_m1} + i_{s\beta_m2}}{2} & \frac{\phi_{ra_m1} + \phi_{ra_m2}}{2} & \frac{\phi_{r\beta_m1} + \phi_{r\beta_m2}}{2} \end{bmatrix}^T \quad (3)$$

$$\omega_e = \frac{\omega_{e_m1} + \omega_{e_m2}}{2} \quad (4)$$

The improved FOC as a function dependent on the engine variables takes the form:

$$FOCI(\hat{i}_{sa}, \hat{i}_{s\beta}, \hat{\phi}_{ra}, \hat{\phi}_{r\beta}, \omega_e) = (v_{sa}^*, v_{s\beta}^*) \quad (5)$$

4.2 Switched Master – Slave Drive Control

The implemented FOC adaptation alternates the necessary inputs from the two estimated and measured engine signals every sampling time period (T_s). This is done using a multiplexed function, which provides the state signals of one engine, every sampling time period. The state variables of the engine considered, usually called “master engine”, are adjusted by the control. Meanwhile, the other one, called “slave engine”, is not considered until the next sampling time period $(k_T + 1)T_s$. Therefore, in the point of view of each engine, the T_s used by the FOC is twice as long as in the previ-

ous control method. Even if the FOC algorithm uses the output of only one rotor flux observer each T_s , the two observers must work continuously. This drive control configuration is shown in Fig. 4. In this case, the function which defines the drive control method becomes:

$$FOC2(\hat{i}_{s\alpha_mi}, \hat{i}_{s\beta_mi}, \hat{\phi}_{ra_mi}, \hat{\phi}_{r\beta_mi}, \omega_{e_mi}) = (v_{s\alpha}^*, v_{s\beta}^*) \quad (6)$$

$$mi = \begin{cases} m1 & \text{if } k_T = \text{odd,} \\ m2 & \text{if } k_T = \text{even.} \end{cases}$$

To PWM VSI

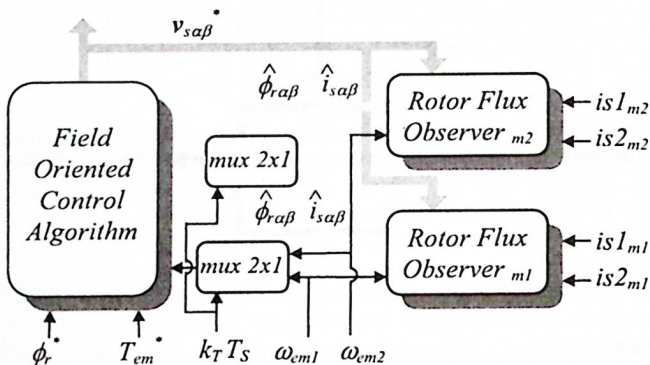


Fig. 4. The Switched Master-Slave drive control structure simplified bloc representation.

4.3 Mean Drive Control for Dual Motor

In this case the induction engines are adjusted by the regulation functions of the classical FOC in the same sampling time period, as it can be seen in Fig. 5. The VSI reference signals obtained at the end of the two regulation processes are averaged, producing the new stator voltage references. A speed engine detector and a rotor flux estimator are essential for the drive control operation of both engines. Yet, the processing time of the resulting algorithm is extended given the necessary in-line computation of two regulation processes. This control method can be represented by the equation:

$$FOC3(FOC_{m1}, FOC_{m2}) = \left(\frac{v_{s\alpha_m1}^* + v_{s\alpha_m2}^*}{2}, \frac{v_{s\beta_m1}^* + v_{s\beta_m2}^*}{2} \right) \quad (7)$$

$$= (v_{s\alpha}^*, v_{s\beta}^*)$$

To PWM VSI

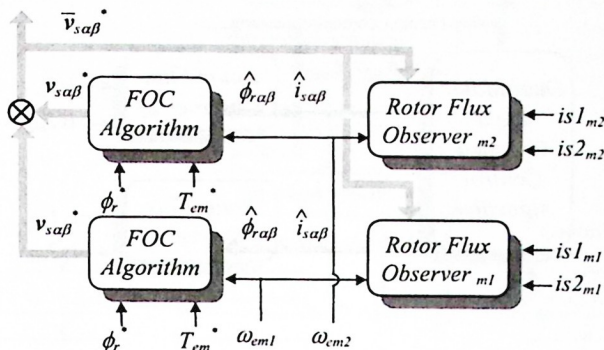


Fig. 5. The Dual Motor Mean Drive control structure simplified bloc representation.

4.4 Mean and Differential Drive Control of Dual Motor

A mean and differential drive control of dual engines has been proposed in [3]. As shown in Fig. 6, it is capable of controlling two different variables by one of the two FOC regulation axes. This is accomplished by the d axis regulation which takes care of the differential engine torque regulation besides that of the rotor flux, as it is the case of the classical FOC. A special transition function between these two is needed for an appropriated system operation.

Equations (8) and (9) define respectively two functions relating the electromagnetic torque and the differential electromagnetic torque produced by the dual engine system. From these equations, it is possible to obtain the expressions of regulation that have been implanted in the dual engine FOC algorithm.

$$\bar{T} = \frac{\bar{T}_{em} - \Delta T_{em} \Delta L_T / \bar{L}_T}{1 - (\Delta L_T / \bar{L}_T)^2} = P \bar{L}_T (\bar{i}_{mr} \times \bar{i}_s + \Delta i_{mr} \times \Delta i_s) \quad (8)$$

$$\Delta T = \frac{\Delta T_{em} - \bar{T}_{em} \Delta L_T / \bar{L}_T}{1 - (\Delta L_T / \bar{L}_T)^2} = P \bar{L}_T (\bar{i}_{mr} \times \Delta i_s + \Delta i_{mr} \times \bar{i}_s) \quad (9)$$

where:

$$\bar{L}_T = \frac{1}{2} \left(\frac{L_{m_m2}^2}{\tau_{r_m2}} + \frac{L_{m_m1}^2}{\tau_{r_m1}} \right), \quad \Delta L_T = \frac{1}{2} \left(\frac{L_{m_m2}^2}{\tau_{r_m2}} - \frac{L_{m_m1}^2}{\tau_{r_m1}} \right).$$

Meanwhile the control structure can be defined from the expression

$$\begin{aligned} &FOC4(\hat{i}_{s\alpha_m1}, \hat{i}_{s\beta_m1}, \hat{\phi}_{r\alpha_m1}, \hat{\phi}_{r\beta_m1}, \omega_{e_m1}, \\ &\hat{i}_{s\alpha_m2}, \hat{i}_{s\beta_m2}, \hat{\phi}_{r\alpha_m2}, \hat{\phi}_{r\beta_m2}, \omega_{e_m2}) = (v_{s\alpha}^*, v_{s\beta}^*) \end{aligned} \quad (10)$$

To PWM VSI

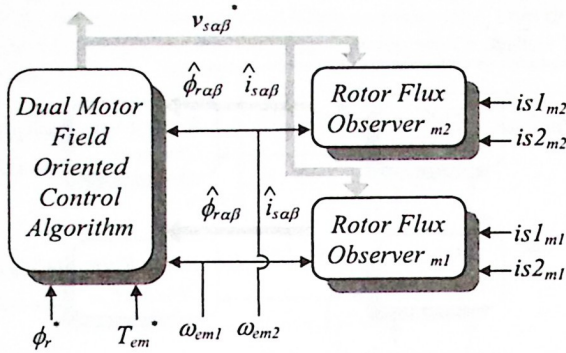


Fig. 6. The Mean and Differential Drive Control for Dual Motor Structure simplified bloc.

The hardware considered for the implementation of the proposed digital-control structures is composed by a DSP-FPGA card, named as PEC-31. It is supported by a measurement card, a protection interface, a distribution rack and a high power bloc. These elements and its interconnectivity are show in the Fig. 7.

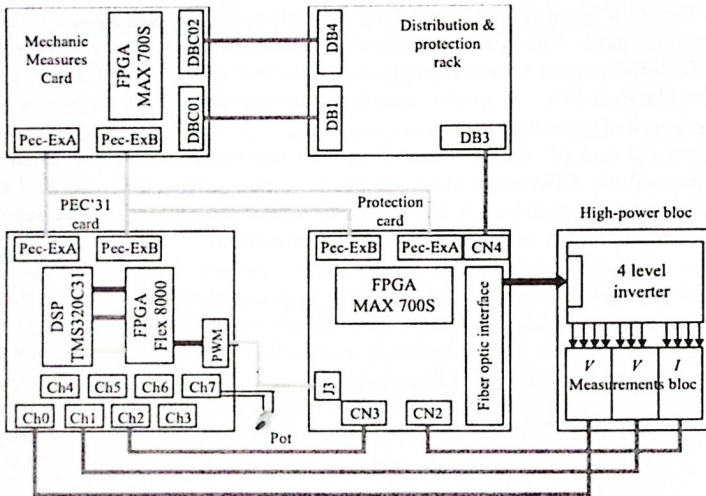


Fig. 7. Hardware for implementation of different control strategies.

5 Software Aspects

The above mentioned control structures have been implemented considering a digital-control circuit based on a DSP-FPGA association. The four digital control algorithms

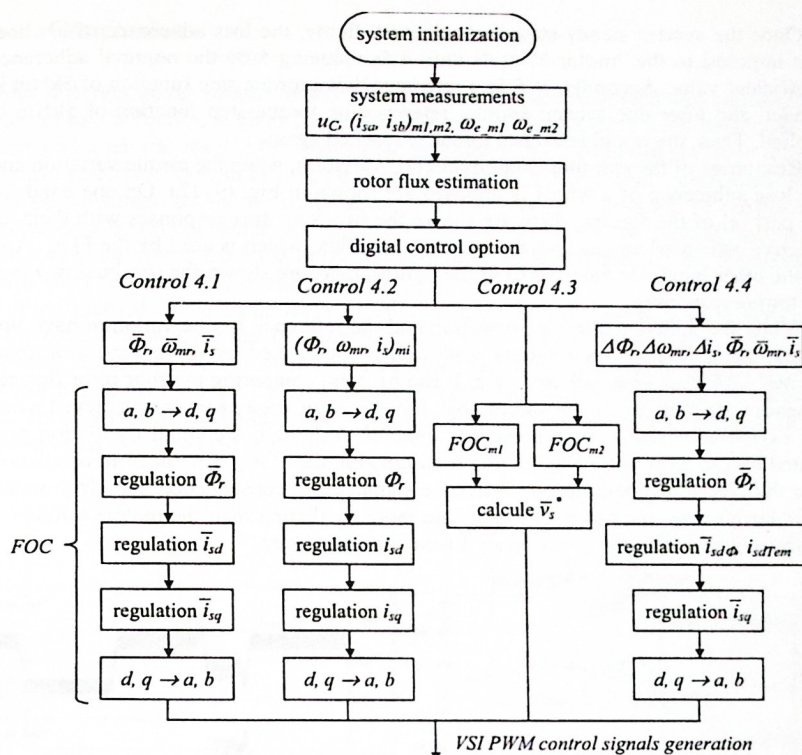


Fig. 8. Implemented algorithm control.

can be included into one main program, as it is shown in Fig. 8. An initial option will define the kind of control to be applied to the high-power system.

Our simulation study has been carried out on two main considerations. First, all modeled electrical and mechanical system components are defined either using SABER library components or created by a MAST program model. And, secondly, that the control system, which includes all the FOC proposed and the rotor flux observer algorithms, are made by a C code program. The objective of such considerations is to achieve a maximum resemblance between the global system model and the real one. The flexibility of our system model permits to incorporate all the estimated aspects in the same control component. Thus, two variables define the number of flux observers to be used and the engine drive control structure to be implemented.

Several simulations had been realized in order to analyze the system performance during imposition of external and internal perturbations. One of the most important perturbations is the loss of adherence of a wheel. Its implications for the railway traction have been explained in several works [2, 6]. Thus, in this paper, this perturbation was imposed after the system had attained the stable-state working conditions defined by: $\phi_r^* = 5 \text{ Wb}$, $T_{em}^* = 10 \text{ kN m}$, and $T_{SS}^* = 7 \text{ m/s}$.

Once the system steady-state had achieved, firstly, the loss adherence of a wheel was imposed to the "motor 1" at 4s until 4.5s reducing 50% the nominal adherence coefficient value. Secondly, at 5.5s a reference down torque step function of 5kNm is applied and after one second another reference up torque step function of 5kNm is applied. Thus, the initial reference torque is reached again.

Responses of the two most important FOC variables, when the torque variation and the loss adherence of a wheel is imposed are shown in Fig. (9–12). On one hand for the part (a) of the figures, there are shown the two rotor flux responses with their respective estimated values and, the average rotor flux, which is used by the FOC. And on the other hand, for the part (b) of the figures, there are shown the obtained two motor torque responses.

When the adherence lost perturbation and the reference torque variation have imposed, the *least perturbed system operation* is established by using the *mean drive control with dual observer structure*. It can be seen, comparing the four rotor flux responses with their estimated values, that, *the best estimation process* is achieved when the *switched master - slave control* is applied. Although, the small oscillation presented by the mean rotor flux, which is assigned to the FOC, introduces an oscillation into the electromagnetic torque reference signal. As a consequence, the electromagnetic torque responses of both motors are more oscillating than the motors torque responses obtained applying the mean drive control structure.

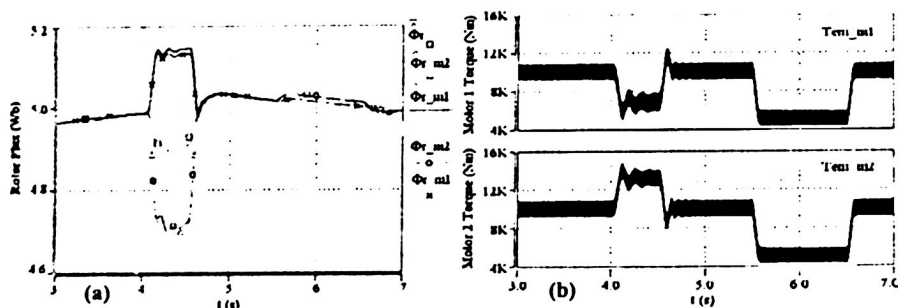


Fig. 9. The Mean Drive Control structure responses of (a) the rotor fluxes and theirs estimation values and, (b) the motor torques.

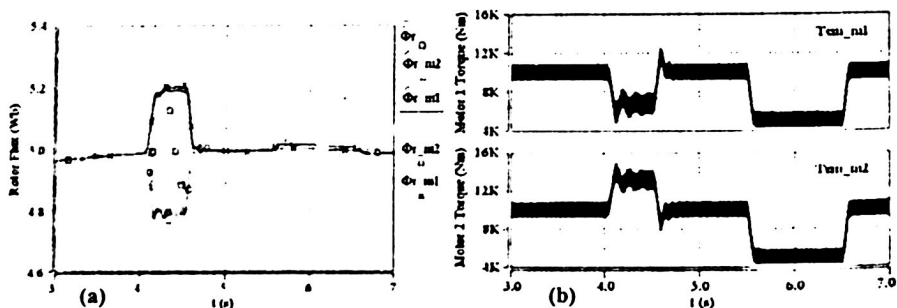


Fig. 10. The Switched Master-Slave Drive Control structure responses of (a) the rotor fluxes and theirs estimation values and, (b) the motor torques.

Once the system steady-state had achieved, firstly, the loss adherence of a wheel was imposed to the "motor 1" at 4s until 4.5s reducing 50% the nominal adherence coefficient value. Secondly, at 5.5s a reference down torque step function of 5kNm is applied and after one second another reference up torque step function of 5kNm is applied. Thus, the initial reference torque is reached again.

Responses of the two most important FOC variables, when the torque variation and the loss adherence of a wheel is imposed are shown in Fig. (9–12). On one hand for the part (a) of the figures, there are shown the two rotor flux responses with their respective estimated values and, the average rotor flux, which is used by the FOC. And on the other hand, for the part (b) of the figures, there are shown the obtained two motor torque responses.

When the adherence lost perturbation and the reference torque variation have imposed, the *least perturbed system operation* is established by using the *mean drive control with dual observer structure*. It can be seen, comparing the four rotor flux responses with their estimated values, that, *the best estimation process* is achieved when the *switched master - slave control is applied*. Although, the small oscillation presented by the mean rotor flux, which is assigned to the FOC, introduces an oscillation into the electromagnetic torque reference signal. As a consequence, the electromagnetic torque responses of both motors are more oscillating than the motors torque responses obtained applying the mean drive control structure.

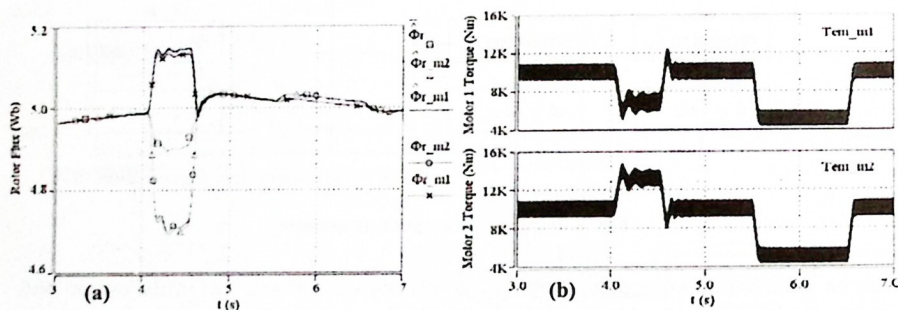


Fig. 9. The Mean Drive Control structure responses of (a) the rotor fluxes and theirs estimation values and, (b) the motor torques.

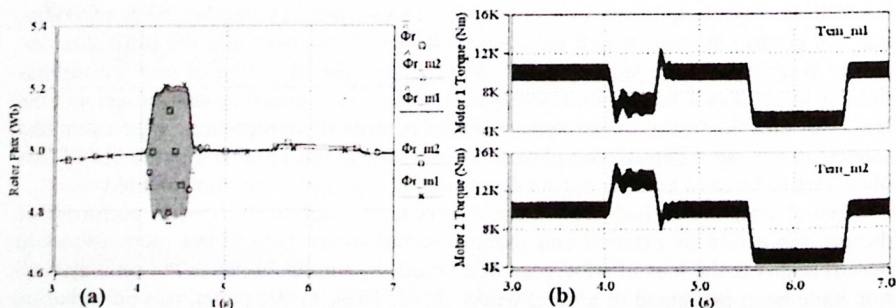


Fig. 10. The Switched Master-Slave Drive Control structure responses of (a) the rotor fluxes and theirs estimation values and, (b) the motor torques.

Switched master-slave control method presents almost the same system behavior as the first control, but the small rotor flux variation is transferred to the other system variables by the FOC algorithm. The mean drive control for dual motor is not really interesting, because its system behavior is not better than any other. In addition, its algorithm is two times longer than the first method.

Nomenclature

C_f	input filter capacitance,	v_s	stator voltage,
E	input voltage,	X	vector or matrix notation,
i_L	input filter current,	Δ	arithmetical mean difference operator,
IM	induction motor,		
i_{mr}	rotor magnetizing current,		
i_s	stator current,		
k_T	positive integer value,		
L_f	input filter inductance,	ϕ_r	rotor flux,
L_m	mutual inductance,	σ	leakage or coupling factor,
L_s	stator self inductance,		$\sigma = 1 - L_m^2 / L_r L_s$
L_r	rotor self inductance,	τ_r	rotor time constant, $\tau_r = L_r / R_r$,
P	pairs of poles number,	τ_s	stator time constant, $\tau_s = L_s / R_s$,
R_f	input filter resistance,	ω	rotating reference frame frequency,
R_r	rotor resistance,		
R_s	stator resistance,	Ω	mechanical speed rotation.
T_{em}	electromagnetic torque,	ω_e	electrical speed rotation;
T_s	sampling time period,		$\omega_e = P\Omega$
TS_{ss}	steady state linear train speed,		
u_C	capacitance voltage,		

Superscripts

•	denotes a reference variable.	^	denotes an estimated value.
•	denotes a time derivative.	—	denotes an averaged value.

Indices

d, q	denote the axis of the rotating reference frame.
α, β	denote the axis of the static reference frame.
$1, 2, 3$	denote the three-phase values.
$-m1, -m2$	denote a variable associated to motor 1 or to motor 2 respectively.
$-M1, -M4$	denote a variable associated to method control 1, ..., method control 4.

References

1. Bouscayrol, A., et al.: Multi-machine multi-converter systems: applications to electromechanical drives in EPJ Applied Physics, 10, 2, (2000) 131-147.

2. Escané, P., Lochot, Ch., Pietrzak-David, M., de Fornel, B. Electromechanical interactions in a high speed railway traction system—Comparison between two drive control structures. CD Rec. EPE 99, Lausanne, Switzerland, 7-9 September, 1999.
3. Kelecý, P. M., Lorenz, R. D. Control Methodology for Single Inverter, Parallel Connected Dual Induction Motors. IEEE PESC'94, Taipei, Taiwan, 20-25 June 1994, 2, 987-991.
4. Provoost, M., Courtois, Ch. Traction électrique ferroviaire, Dynamique ferroviaire et sous-stations. Techniques de l'Ingénieur, traité Génie électrique, D 5 501, Paris, France, (1997).
5. Peña-Eguiluz, R., Pietrzak-David, M., de Fornel, B. Observation Strategy in a Mean Control Structure for Parallel Connected Dual Induction Motors in a Railway Traction Drive System. CD Rec. EPE 2001, Graz, Austria, 27-29 August 2001.
6. Peña-Eguiluz, R., Pietrzak-David, M., Roboam, X., de Fornel, B. Dead time effect in a Railway Traction System. Proc. IEEE ISIE 2000, Puebla, Mexico, 2000, 1, 151-156.

Author Index

Índice de autores

- Acevedo Mosqueda, Marco A. 41
 Aldeco Pérez, Rocío A. 149
 Alor Hernández, Giner 171
 Angeles-Yreta, A. 127
 Bernábe Loranca, Beatriz 191
 Boone Rojas, Ma. del Rocío 191
 Brena, Ramon F. 3
 Bueno-Tecpanecatí, Claudia 83
 Burlak Gennadiy, N. 23
 Cabrera, Rubí J. 75
 de Fornel, Bernard 227
 Durán Camarillo, Edmundo René 51
 Esquivel Loria, Br. Zoemy 213
 Figueroa-Nazuno, J. 101, 127
 Gallegos Funes, Francisco J. 35, 41
 González Dueñas, Carlos Alberto 51
 González Montiel, Aranzazú 51
 Guzmán, Giovanni. 203
 Heredia López, Francisco J. 213
 Hernández, Erika 91
 Jiménez-Salazar, Héctor 75, 83, 91
 Khachaturov, Vladimir 111
 Landassuri-Moreno, V. 101
 Lavalle, Jesús 91
 Leal Bando, Lorena 63
 León Chávez, Miguel A. 137, 149
 López Sales, Ramiro 191
 López y López, Fabiola 63, 159
 Macías Galindo, Daniel 159
 Martínez, Emmanuel 3
 Martínez Mora, Verónica 51
 Martínez Valdes, Jesús 41
 Mendoza González, Carlos Vidal 51
 Merino Higareda, Sugehi M. 137
 Montiel Pérez, Jesús Yaljá 13
 Moreno, M. 203
 Olmedo Aguirre, José Oscar 171
 Páris, Jehan-François 179
 Peña-Eguiluz, Rosendo 227
 Pietrzak-David, M. 227
 Pinto, David 75, 83, 91
 Khachaturov, Volodymyr 111
 Quintero, Rolando. 203
 Romero Herrera, Rodolfo 13
 Serrano Talamantes, Félix 13
 Starostenko, Oleg 23
 Tomás Tello, José 23
 Torres, Miguel. 203
 Torres Manriquez, Juan José 51
 Varela-Benítez, José Luis 35
 Vilariño Ayala, Darnes 63, 75
 Zavala Díaz, José Crispin 111

Advances in Computer Science in México
Impreso en los Talleres Gráficos
de la Dirección de Publicaciones del
Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, DF.
Mayo 2005. Edición: 500 ejemplares.

1

Since millennial calendars and perfect geometry of the pyramids, since the first computer 47 years ago, since foundation of the first Computing Research Center 7 years ago, computing in Mexico has rich history, solid tradition, excellent present, and brilliant future.

This volume contains 22 carefully selected papers by 52 authors, who are, in accordance with the theme of this volume, mainly from Mexico, but also from USA and Europe. The papers are arranged into five groups:

Robotics and vision, including such topics as planning, mobile robots, motion prediction, noise cancellation filters, wavelets analysis, and visual authentication;

Natural Language Processing, including such topics as information retrieval, information extraction, and software infrastructure for natural language processing;

Algorithms, including such topics as time series forecasting with neural networks, discrete optimization, and object identification using neural networks;

Software Technologies and Databases, including such topics as cryptography, agent technology, web services, video-on-demand optimization, database integrity, and spatial databases;

Hardware and Control, including such topics as telemetric systems and digital-based control of engines.

ISSN: 1665-9899

INSTITUTO POLITÉCNICO NACIONAL

