Advances in Soft Computing Algorithms

Research in Computing Science

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 49** November, 2010. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor Responsable: *Juan Humberto Sossa Azuela, RFC SOAJ560723*

***Research in Computing Science** is published by the Center for Computing Research of IPN. **Volume 49**, November, 2010. 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, November, 2010, in the IPN Graphic Workshop – Publication Office.*

Volume 49

Volumen 49

Advances in Soft Computing Algorithms

Volume Editors:
Editores del Volumen

Grigori Sidorov
Arturo Hernández-Aguirre
Carlos Alberto Reyes-García

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2010

ISSN: 1870-4069

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

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

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

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

The purpose of this volume is to reflect the new directions of investigation in the areas of Computer Science related to Artificial Intelligence (AI), and more specifically, this issue is focused on algorithms that are based on AI in different ways.

Artificial Intelligence is a branch of computer science that models human ability of reasoning, usage of human language and organization of knowledge, solving problems and practically all other human intellectual abilities. Usually it is characterized by application of heuristic methods because in the majority of cases there is no exact solution to this kind of problems.

Papers for this volume were carefully selected by volume editors on the basis of the blind reviewing process performed by editorial board members and additional reviewers. The main criteria for selection were their originality and technical quality.

This issue of the journal Research in Computing Science can be interesting for researchers and students in computer science, especially in areas related to artificial intelligence, and also for persons who are interested in cutting edge themes of the computer science. Each submission was reviewed by three independent members of the editorial board of the volume or additional reviewers.

This volume contains revised versions of 14 accepted papers. The papers are structured into the following six sections:

- Naturally Inspired Algorithms,
- Specialized Algorithms,
- Data Mining Algorithms,
- Robotics,
- Fuzzy Logic Algorithms,
- Natural Language Processing.

This volume is a result of work of many people. In the first place, we thank the authors of the papers included in this volume for the technical excellence of their papers that assures the high quality of this publication. We also thank the members of the International Editorial Board of the volume and the additional reviewers for their hard work consisting in selection of the best papers out of many submissions that were received.

The submission, reviewing, and selection process was performed on the basis of the free system EasyChair, www.EasyChair.org.

November, 2010

Grigori Sidorov,
Arturo Hernández-Aguirre,
Carlos Alberto Reyes-García

Table of Contents
Índice

Page/Pág.

Naturally Inspired Algorithms

Specialized Algorithms

Robotics

Data Mining Algorithms

Fuzzy Logic Algorithms

Natural Language Processing

Naturally Inspired Algorithms

Optimization of a Reactive Control in a Mobile Robot using Evolutionary Algorithms and Fuzzy Logic

Abraham Meléndez, Oscar Castillo, and Arnulfo Alanis

Tijuana Institute of Technology,
Tijuana, Mexico
abraham.ms@gmail.com,ocastillo@hafsamx.org

Abstract. This paper describes an evolutionary algorithm applied to the optimization of a reactive controller applied to a mobile robot. The Genetic algorithm will optimize the Fuzzy Inference System's membership parameters and fuzzy rules.

Keywords: Mobile Robot, Fuzzy Logic, Evolutionary Algorithms.

1 Introduction

Robots are being more commonly used in many areas of research and a reason for this is that they are becoming more accessible economically for researchers. In this paper we consider the optimization of a fuzzy controller; that gives the ability of reaction to the robot. This may be too general so let's limit what in this paper will be described as ability of reaction, this is applied in the navigation concept, so what this means is that when the robot is moving and at some point of its journey it encounters an unexpected obstacle it will react to this stimulation avoiding and continuing on its path. The trajectory and path following are considered independent parts and are not consider on this paper [19].

There are many traditional techniques available to use in control, such as PD, PID and many more but we took a different approach in the Control of the robot, using an area of soft computing which is fuzzy logic that was introduced by Zadeh [1]. Later this idea was applied in the area of control by Mandami [2] where the concept of FLC (Fuzzy Logic Controller) originated. It's also important to mention that this is not the only area were the fuzzy concepts are applied but it's where the most work has been done, and were many people have contributed important ideas and methods like Takagi and Sugeno [2].

This paper is organized as follows: In section 2 we describe the mobile robot used in these experiments, section 3 describes the development of the evolutionary method. Section 4 shows the simulation results. Finally section 5 shows the Conclusions.

2 Mobile Robot

In this section we describe the particular mobile robot considered in this work. The robot is based on the description of the the Simulation toolbox for mobile robots[22] assumes wheeled mobile robot consisting of one or two conventional, steered, unactuated and not-sensed wheels and two conventional, actuated, and sensed wheels (conventional wheel chair model). This type of chassis provides two DOF (degrees of freedom) locomotion by two actuated conventional non-steered wheels and one or two unactuated steered wheels. The Robot has two degrees of freedom (DOFs): y-translation and either x-translation or z-rotation [22], Fig. 1 shows the robots configuration, it will have 2 independent motors located on each side of the robot and one castor wheel for support located at the form of the robot.

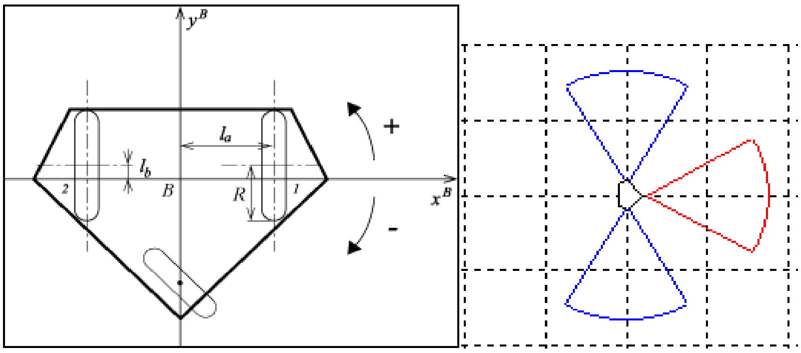


Fig. 1. Kinematic coordinate system assignments [22].

The kinematic equations of the mobile robot are as follows:
Equation 1 is the sensed forward velocity solution [22]

$$\begin{pmatrix} v_{Bx} \\ v_{By} \\ \omega_{Bz} \end{pmatrix} = \frac{R}{2l_a} \begin{pmatrix} -l_b & l_b \\ -l_a & -l_a \\ -1 & 1 \end{pmatrix} \begin{pmatrix} \omega_{W1} \\ \omega_{W2} \end{pmatrix} \tag{1}$$

Equation 2 is the Actuated Inverse Velocity Solution [22]

$$\begin{pmatrix} \omega_{W1} \\ \omega_{W2} \end{pmatrix} = \frac{1}{R(l_b^2+1)} \begin{pmatrix} -l_a l_b & -l_b^2-1 & -l_a \\ l_a l_b & -l_b^2-1 & l_a \end{pmatrix} \begin{pmatrix} v_{Bx} \\ v_{By} \\ \omega_{Bz} \end{pmatrix} \tag{2}$$

Where under the Metric system.

V_{Bx}, V_{By} Translational velocities [$\frac{m}{s}$],

ω_{B_z} Robot z-rotational velocity [$\frac{rad}{s}$],

ω_{W_1}, ω_{W_2} Wheel rotational velocities [$\frac{rad}{s}$],

R Actuated wheel radius[m],

l_a , l_b Distances of wheels from robot's axes [m].

3 Evolutionary Method Description

In this section we will cover the Genetic Algorithm applied to our problem of finding the best fuzzy reactive controller for a mobile robot.

The purpose of using a evolutionary method, is to obtained the best reactive control possible, for the robot, but also taking into consideration other desirable characteristics on the robot that we want to improve making this a multi objective [17] problem, for this we will take advantage of the HGA (Hierarchy Genetic Algorithm) intrinsic characteristic to solve multi objective problems, now let us state the main goal of our HGA.

The main goal is to optimize the Reactive Control taken in to consideration the following:

- Fine tune the Fuzzy Memberships,
- Optimize the FIS if then fuzzy rules,
- The mobile robot Power Usage.

In Fig. 4, we show the global cycle process of the GA, under the Evaluation of the each individual, is where we are going, to measure the goodness of each of the FIS (Fuzzy Inference System) represent by each Individual chromosome, in our test area, that will take place in a unknown environment (Maze [19]) to the robot where the robot´s objective will be find the exit, avoiding hitting the walls and any other obstacle present.

Our criteria to measure the FIS global performance will take into consideration the following:

- Cover Distance,
- Time use to cover distance,
- Battery life.

All of these variables are the inputs of the Evaluation FIS that we will use to obtain the fitness value of each chromosome.

The FIS that we optimize is a Mamdani type fuzzy system, consisting of 3 inputs that are the distances obtain by the robots sensors describe on section 2, and 2 outputs that control de velocity of the servo motors on the robot, all this information is encoded on each chromosome.

The chromosome architecture, shown on Fig. 4, where we have encoded the membership functions type and parameters, we have set a maximum number of 5 membership functions for each of the outputs and input and output variables.

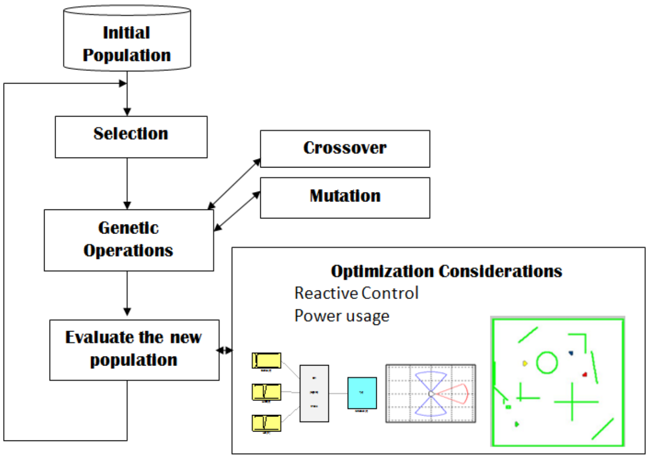


Fig. 2. Genetic Algorithm process.

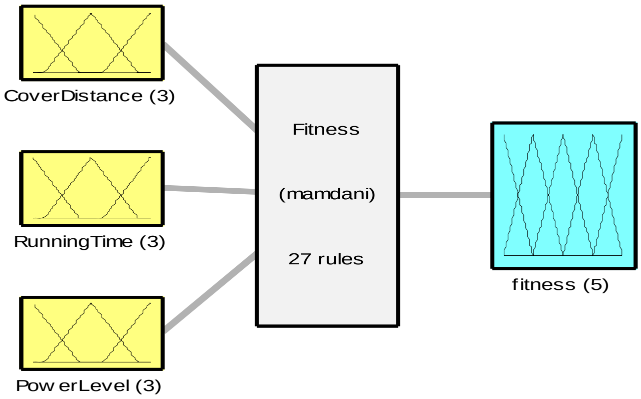


Fig. 3. Fitness FIS.

All the results obtain will get persisted on a Data Base, were we will store each step on the genetic cycle, keeping track of the genealogy of each chromosome, and with this we can examine each of the top individuals and can back track the behavior of the genetic algorithm.

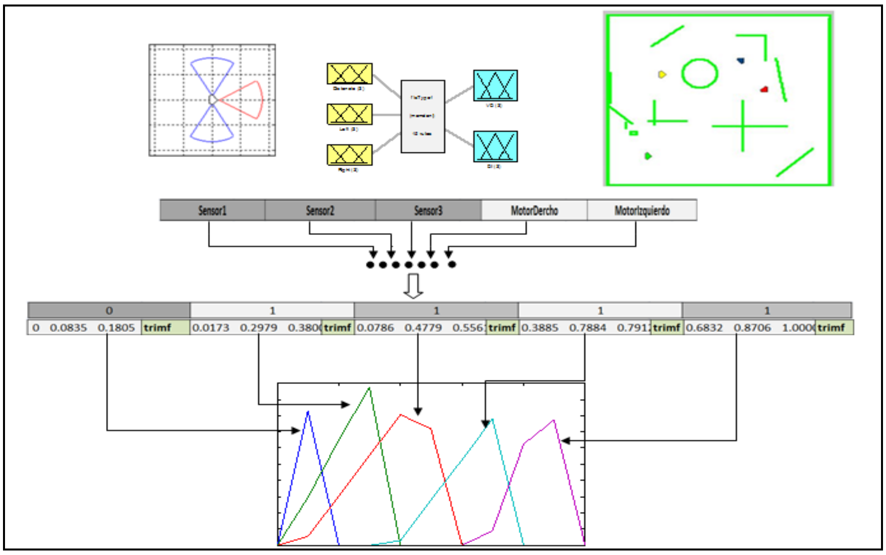

Sensor1
Sensor2
Sensor3
MotorDercho
MotorIzquierdo
0
1
1
1
1
0 0.0835 0.1805 trimf
0.0173 0.2979 0.380(trimf
0.0786 0.4779 0.556: trimf
0.3885 0.7884 0.791: trimf
0.6832 0.8706 1.000(trimf

Fig. 4. Chromosome Architecture.

4 Simulation Results

We have worked on the reactive control for a mobile robot before, where we use a particular maze problem to test the effectiveness of each of the reactive controls, and we did not use any optimization strategy to fine tune the controllers as it was a manually process, and from that experience is that we decided to apply GA to this problem. On the Fig. 5 and Table 1 you can see the results we obtained in our prior experiments.

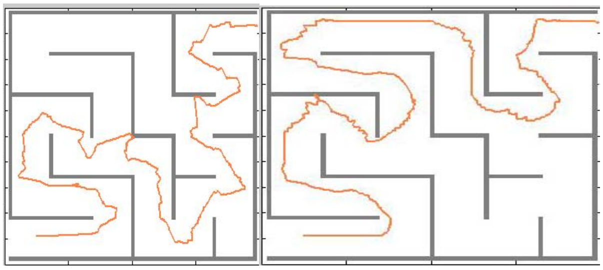


Fig. 5. Chromosome Architecture.

Table 1. Reactive Control Non Optimized Results.

Experiment	LAS	RAS	Time	FE
27 Rules FIS				
9	38.26	38.46	60.34	Yes
10	40.42	40.64	59.50	Yes
12	40.06	40.12	60.70	Yes
Average	*39.58*	*39.74*	*60.18*	
Standard Deviation	*1.16*	*1.14*	*0.62*	
10 Rules FIS				
1	31.51	47.43	55.75	Yes
2	35.55	54.33	51.80	Yes
3	36.66	53.68	55.40	Yes
4	36.66	53.68	54.10	Yes
5	36.91	54.18	49.95	Yes
6	35.44	52.51	46.65	Yes
7	36.14	51.66	49.85	Yes
8	33.48	49.58	51.76	Yes
9	37.61	51.51	55.15	Yes
Average	*35.55*	*52.06*	*52.27*	
Standard Deviation	*1.92*	*2.32*	*3.10*	

*LAS=Left Motor Average Speed
*RAS= Right Motor Average Speed
*FE= Found Exit

5 Conclusions

At the stage of this paper we are on the experimenting phase, our first result show promising data were we expect that the HGA will improve the overall performance of the mobile robot, and improve the results obtained previously. We will test our best reactive controller obtained with the HGA under the same maze problem this in order to establish a comparison between the controllers.

Acknowledgment. We would like to express our gratitude to the CONACYT, and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

1. Aceves, A., Aguilar, J.: A Simplified Version of Mamdani's Fuzzy Controller: The Natural Logic Controller. IEEE Transactions on Fuzzy Systems, Vol. 14, No 1, 16-30 (2006)
2. Astudillo, L., Castillo, O., Aguilar L.T.: Intelligent Control of an Autonomous Mobile Robot Using Type-2 Fuzzy Logic. Journal of Nonlinear Studies, Vol. 14, No. 1., 37-48
3. Astudillo L., Castillo O., Aguilar L.: Control Difuso de Robots Autónomos Móviles en Ambientes Inciertos usando Lógica Difusa. Tesis, División de Estudios y Posgrados e Investigación, ITT, México (2006).
4. Campion G., Bastin G., D'Andrea-Novel B.: Structural Properties and Clasification of Kinematic and Dynamic Models of Wheeled Mobile Robots. IEEE Trans. On Robotics and Automation, Vol. 12, No 1, 47-62 (1996)
5. Cardenas S., Castillo O., Aguilar L.: Controlador de Seguimiento para un Robot Móvil empleando Lógica Difusa. Tesis, División de Estudios y Posgrados e Investigación, ITT, México (2005).
6. Conde Bento L., Nunes Urbano, Mendes A., Parent M.: Path-Tracking Controller of a bi-steerable Cybernetic Car using Fuzzy Logic. In: Proceedings of ICAR 2003, The 11th International Conference on Advanced Robotics. pp. 1556-1561 (2003)
7. Cupertino, F., Giordano, V., Naso, D., Delfine, L.: Fuzzy control of a mobile robot. Robotics & Automation Magazine, IEEE, pp. 74-81.
8. Erkkinen, T.: Embedded Coder Robot NXT. http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=13399.
9. Erkkinen, T.: Embedded Coder Robot for LEGO® Mindstorms® NXT Rev. 3.03. http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=13399.
10. Ishikawa S.: A Method of Indoor Mobile Robot Navigation by Fuzzy Control. In: Proc. Int. Conf. Intell. Robot. Syst., Osaka, Japan, pp. 1013-1018 (1991)
11. Thomson, A., Baltes, J.: A path following system for autonomous robots with minimal computing power. University of Auckland. Private Bag 92019. Auckland, New Zealand. Technical Report (2001)
12. Jang, J.-S. R., Sun, C.-T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall Chapter (1997)

14 Kulkarni A.: Computer Vision and Fuzzy-Neural Systems. Prentice Hall PTR (2001)

15 Leyden, M, Toal D., Flanagan, C.: A Fuzzy Logic Based Navigation System for a Mobile Robot. In: Proceedings of Automatisierungs Symposium, Wismar, Germany (1999)

16 Macek, K., Petrovic, I., Siegwart, R.: A control method for stable and smooth path following of mobile robots. In: Proceedings of the 2nd European Conference on Mobile Robots - ECMR 2005, Ancona, Italy, pp. 128-133 (2005)

17 Man. K. F.: Genetic Algorithms Concepts and Designs (Advanced Textbooks in Control and Signal Processing). Springer, Corrected edition (1999)

18 Martínez R., Castillo O., Aguilar L.: Control Inteligente de Robots Autónomos Móviles Bajo Pares Perturbados Usando Lógica Difusa Tipo-2. Tesis, División de Estudios y Posgrados e Investigación, ITT, México (2008)

19 Meléndez, A., Castillo, O., Soria, J.: Reactive Control of a Mobile Robot in a Distributed Environment Using Fuzzy Logic. In: Fuzzy Information Processing Society, NAFIPS 2008. Annual Meeting of the North American Vol. 19-22, pp. 1-5 (2008)

20 Payton, D.W., Rosenblatt, J.K., Keirsey, D.M.: Plan guided reaction. IEEE Transactions on Systems, Man and Cybernetics, Volume 20, Issue 6, pp. 1370–1382

21 Peri, V.M., Simon, D.: Fuzzy logic control for an autonomous robot. In: NAFIPS 2005. Annual Meeting of the North American Fuzzy Information Processing Society, pp. 337-342 (2005)

22 Pishkenari H. N., Mahboobi S. H., Meghdari A.: On the Optimum Design of Fuzzy Logic Controller for Trajectory Tracking Using Evolutionary Algorithms. In: IEEE Conference on Cybernetics and Intelligent Systems, vol. 1, pp.660-665 (2004)

Objects Detection and Tracking using Evolutionary Algorithms

Mónica Pérez-Meza[1], Rodrigo Montúfar-Chaveznava[2],
and Alberto Jaramillo-Núñez[3]

[1] Universidad de la Sierra Sur, Guillermo Rojas Mijangos s/n,
Ciudad Universitaria, 70800 Miahuatlán de Porfirio Díaz, Oaxaca, Mexico
[2] Instituto de Ciencia y Tecnología, República de Chile 6, Centro, 06010,
Del. Cuauhtémoc, D.F., Mexico
[3] Intituto Nacional de Astrofísica, Óptica y Electrónica,
Apartado Postal 51 y 216, Puebla, 72000, Mexico
mperez@hotmail.com, rodrigo_montufar@hotmail.com, ajaramil@inaoep.mx

Abstract. The aptitude to detect objects is a present and predominant characteristic in the vision systems of the live beings. This characteristic is so important that it has been a topic of investigation in the area of artificial vision. In this paper we link 3D scenes reconstruction to evolutionary algorithms through the stereo vision that usually needs of two images captured by a pair of cameras, in order to determine the distance of the objects in relation to a reference system. From this is obtained the third dimension of the objects. In this work we propose to use only a camera, which is displaced along a path, capturing images every certain distance. As we cannot perform all computations required for the total scene reconstruction, we employ an evolutionary algorithm to partially reconstruct the scene and obtain its representation. The algorithm employed is the fly algorithm, which employ spatial points named "flies" to reconstruct the most important characteristics of the world. Flies (points in the space) are evaluated by a fitness function; the flies with better fitness are posed on the object to be detected. Finally, the center of mass of the flies with the highest fitness is obtained and a geometric form is recuperated using the parameters of the centroid, this form will be use to track the object.

Keywords: 3D Reconstruction, Artificial Vision, Stereo Vision, Evolutionary Algorithms.

1 Introduction

In this work we present a method to carry out the objects detection and tracking of a dynamic scene using stereoscopy, evolutionary algorithms and monocular vision.

Some algorithms are suitable to give solution to the stereo vision problem such as the evolutionary algorithms. In this case, we employ the fly algorithm with some modifications according to the conditions, tasks and proposed objectives.

Stereo vision is employed to obtain the depth of the objects present in a scene. The capture of slightly displaced images of a scene can be carried out by one of the following procedures:

Static capture. Aligning two or more cameras separated among them (similar to mammalian vision system), every camera captures one image.

Dynamic capture. Displacing one camera along a path and taking images in different positions of the path.

There are works where two or more cameras are used to obtain the reconstruction of a scene as in [4,9]. As was mentioned above, in this work we propose to use one camera and a dynamic capture. After the stereo images are obtained, we performed the 3D scene reconstruction applying projective geometry and using the fly algorithm to keep the best points in a scene of the object to be detected. These points are flies posed on the object, and with these points we obtain the centroid for tracking.

2 Stereo Vision

Stereo vision is a method for obtaining the three-dimensional view of the objects in a scene. It is determined by the distance of objects in relation with a reference system.

In stereo vision, we need two or more cameras slightly displaced one of another, for obtaining the images. The pair of images has many characteristics in common, but they also have certain differences, and these differences are called disparity [10].

3 Evolutionary Algorithms

Evolutionary algorithms manipulate individuals which are evaluated by a fitness function, in analogy to the biological evolution. The general schema of evolutionary algorithms is shown in Fig. 1.

The principal characteristics are:

- The population is a group of individuals.
- An individual is defined by genes $X = (x_1, x_2\ldots x_n)^T$, which particularly represents its position (X, Y, Z) in the space.
- The evaluation is the computation of the fitness value in every individual.
- The selection eliminates part of the population, keeping the best individuals.
- The evolution applies genetic operators (crossover, mutation, etc.), leading to generate new individuals in the population.
- Some kinds of evolutionary algorithms are:
- Genetic algorithms, which are a technique of programming that imitate the biological evolution as a strategy to solve problems.
- Evolutionary strategies, which are rules that define the behavior of the individual under certain circumstances.
- Genetic programming, which are specific instructions in a programming language.

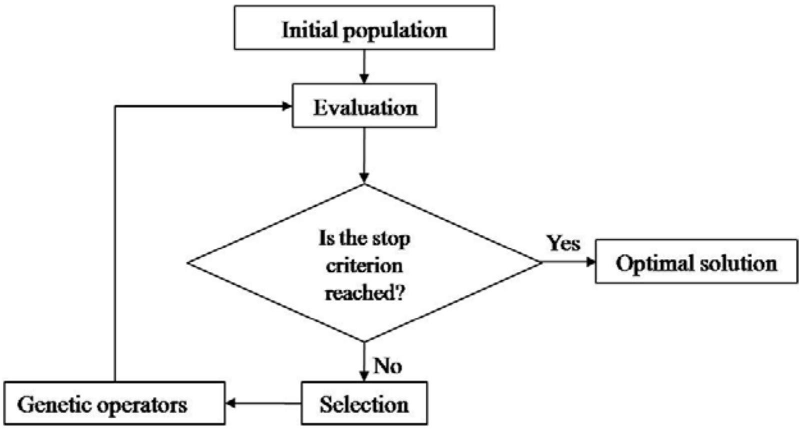


Fig. 1. Schematic description of the Genetic Algorithms method.

In this work we employ genetic algorithms. These algorithms evolve a population of individuals submitting it to random actions, like in the biological evolution, and to a selection process according to certain criterion, where the most adapted individuals are selected to survive the process and the less adapted are ruled out.

3.1 The Fly Algorithm

The fly algorithm is considered an image processing technique based on the evolution of a population of flies (points in the space) projected over stereo images. The evolution is regulated by a fitness function determined in such a way that the flies converge on the surface of an object located in the scene.

A fly is defined as a 3D point with coordinates (x, y, z). The flies are projected over a couple of despaired images by stereoscopy, producing a pair of 2D coordinates: (x_R, y_R) and (x_L, y_L) for the right and left images, respectively.

Initially, the population of flies is generated randomly in the intersection area of the view of both images and equally distributed on a certain number of regions to disperse all of them.

The fly algorithm considers the following functions:

The fitness function. The fitness function evaluates a fly, comparing projections on the images. If a fly is located on an object, the projections will have similar pixel neighborhoods on both images and the fly will have a high fitness value. This idea is illustrated in Figs. 2 and 3. Figure 3 shows the neighborhoods of two flies on left and right images. In this example, fly 1, which is located on an object, has a better fitness value than fly 2. The fitness function F defined in [9,11] is:

$$F = \frac{|\nabla(M_L)|.\,|\nabla(M_R)|}{\left(\sum_{colors}\sum_{(i,j)\in N}[L(x_L+i,y_L+j)-R(x_R+i,y_R+j)]^2\right)} \tag{1}$$

$$G = |\nabla(M_L)|.|\nabla(M_R)| \qquad (2)$$

where:

- (xL, yL) and (xR, yR) are the pixel coordinates of the fly projected on the left and right images, respectively.
- L(xL + i, yL + j) and R(xR + i, yR + j) are the color values of the pixels in a neighborhood at left and right images, respectively.
- N is the dimension of the neighborhood population introduced to obtain a more discriminating comparison of the fly projections.
- $|\nabla(M_L)|$ and $|\nabla(M_R)|$ are the norms of the gradients of Sobel on the left and right projections of the fly. That is intended to penalize flies when they are located on uniform regions.
- In color images, the difference of squares in (2) on each color channel is computed. In grayscale images is computed for one channel.

Selection. Selection is elitist and deterministic. It classifies flies according their fitness values and keeps the best individuals. A sharing operator [4, 9] reduces the fitness of clustered flies packed and forces them to explore other areas on the world.

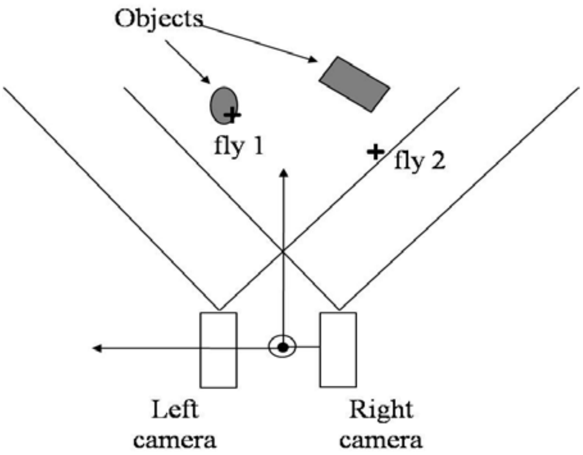


Fig. 2. View of two flies in the scene. The flies are projected on left and right image plans.

Genetic operators. We apply a pair of genetic operators to selected individuals.
Barycentric cross-over. A new individual is generated from two parents F_1 and F_2, this individual is positioned between them, as follow:

$$\overrightarrow{F_3} = \lambda\overrightarrow{F_1} + (1 - \lambda)\overrightarrow{F_2} \qquad (3)$$

where λ is a random value uniformly distributed between [0, 1].
Gaussian mutation. A new fly is generated adding Gaussian noise to each coordinate of the parent fly.

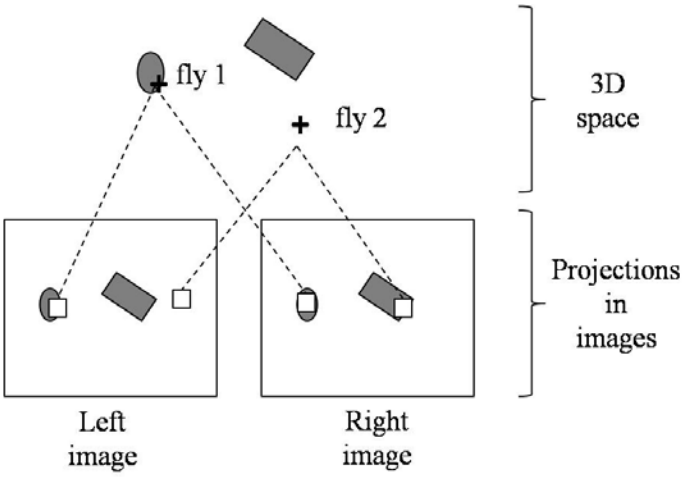


Fig. 3. Flies projections on the left and right images.

4 Image Processing

We have incorporated an image processing operator to enhance the performance of the fly algorithm and the visualization of the results. The process used was the edge detection.

4.1 Edge Detection

Edge detection is used to locate points where a sharp intensity variation is presented. The basic solution for many edge detection algorithms is the computation of local differential operators. We employ the Sobel operator [10] in the fitness function.

The Sobel operator measures the 2D spatial gradient on an image, emphasizing the regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in a grayscale image.

The Sobel operator consists of a pair of 3×3 convolution kernels. One kernel is simply the other rotated by 90°. This is very similar to the Roberts Cross operator [10].

5 Objects Detection and Tracking

5.1 Objects Detection

Objects recognition in an image labels sections of the image, given certain characteristics. This recognition is separated into two stages which are the detection and classification of the object.

To objects detect, we perform the edge and contours detection of the image. Each contour is considered an object to classify. For the classification, we transform the 2D representation of the contour to a radial representation that can be taken to a dimension. This representation will be made starting from the "center of mass" of the contour.

For this work, the center of mass is obtained using a proportion of flies with a high fitness value.

5.2 Object Tracking

The object tracking is calculated with the position and shape of the object in each image of a sequence of images. We can determine the motion of an object by the position and shape over a period of time.

To track a moving object through a set of images it is necessary to use the points resulting from object detection in images and to calculate the proximity of the points of the object in subsequent images.

The representation that is used in objects tracking can be from a point on the object to the 3D representation, through forms or appearances. Some forms are: points, geometric shapes, contours or shapes, articulated forms, skeletons, and so on. The representation we use in this work in objects tracking is a point which is located in the center of mass of the object.

6 Detection and Tracking System

The developed system for detection and tracking is showed in Fig. 4. It has four basic modules: Obtaining images, Reconstruction, Object detection and Object tracking. The Reconstruction module, shown in Fig. 5, has five sub-modules: Fitness function, Best flies, Crossing, Mutation and Correspondence.

The modules are:

a) Obtaining images. It obtains images (stereo pair), slightly five centimeters displaced one from the other, and these images are used in the Reconstruction module.
b) Object detection. It detects the object using a group of high fitness flies, the flies are located on the object.

c) Object tracking. After the object has been detected, its centroid is obtained with this module using the flies of high fitness.

d) Reconstruction. It performs the 3D reconstruction from the sequence of images. It has the following sub-modules.

   i) Fitness function. The fitness function is applied to the population of flies. If the color differs between each fly and its neighborhood are low, almost zero, the value of the fitness function is high for this fly and vice versa.

   ii) Best flies. In this module we obtain the flies with the highest fitness.

   iii) Crossing. This module generates new flies from two parents using the crossing function.

   iv) Mutation. In this module new flies are obtained by adding Gaussian noise to parent flies.

   v) Correspondence. In this module we obtain the correspondence of the flies projected on the right and left images by stereoscopy.

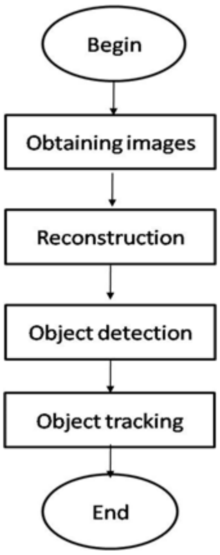


Fig. 4. Flow diagram of objects detection and tracking system.

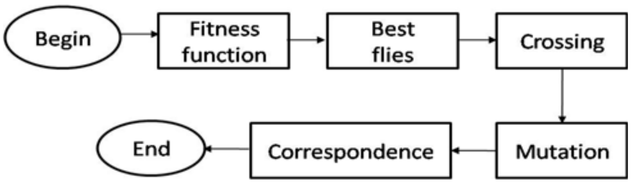


Fig. 5. Flow diagram of Reconstruction module.

Figure 5 shows the flow diagram of the Reconstruction module, where the fitness function has been computed for all individuals. We keep the best flies and the worst flies are eliminated when the operators of mutation and crossing are applied. These operators are applied at different percentages to the previous population to be projected on the right and left images. We obtain the centroid for detection and tracking with the flies with high fitness, these flies are located on the object.

7 Results

We present the results obtained using a population of 3000 flies, a set of images displaced two inches one from another on a horizontal axis. The left and right images are refreshed every 1.5 seconds approximately. In this exercise we obtain the reconstruction in the following way: the right image substitutes the left image and a new image is placed in the location corresponding to the right image. Next, with the best flies we obtain the centroid and so, we perform the object tracking. This process is showed in Fig. 6. The parameters employed in the fly algorithm are:

- Flies preserved every generation: 35%,
- Flies generated by crossing: 5%,
- Flies generated by mutation: 50%,
- Flies randomly generated: 10%.

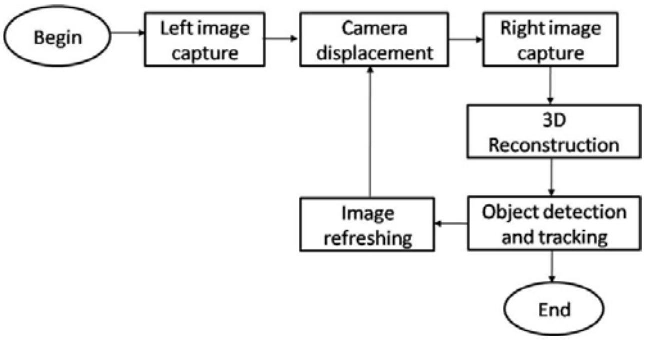
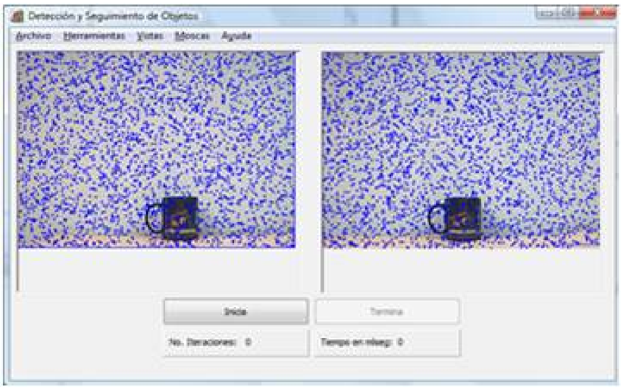


Fig. 6. Schematic description of the objects reconstruction, detection and tracking.

Figure 7 shows the left and right images. The initial population of flies is also presented. After 30 iterations, the best flies are in red, located on the object and the rest of the population is in blue. The centroid is obtained to perform object detection and tracking, as presented in Fig. 8. Next, the camera is translated two inches on the horizontal axis, then the left image is discarded, the right image takes left image place and a new image is placed in the location corresponding to the right image. This new disposition is shown in Fig. 9. We can observe that neither the flies nor the centroid are placed on the object.

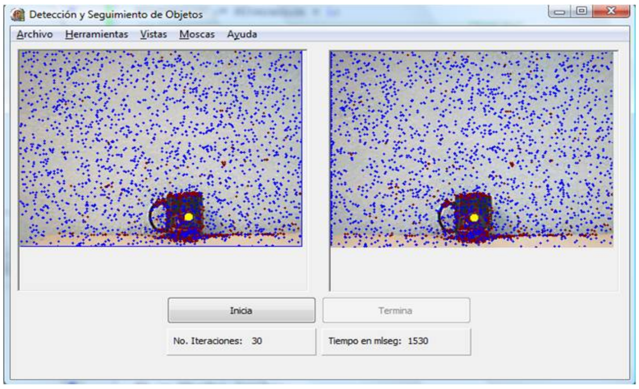
Figures 10 to 12 show the results of the second, third and fourth pairs of images after 59, 89 and 117 executions. The object is detected and tracked; the flies are posed on the objects again.


Detección y Seguimiento de Objetos
Archivo Herramientas Vistas Moscas Ayuda
Inicia
Termina

No. Iteraciones: 0

Tiempo en mlseg: 0

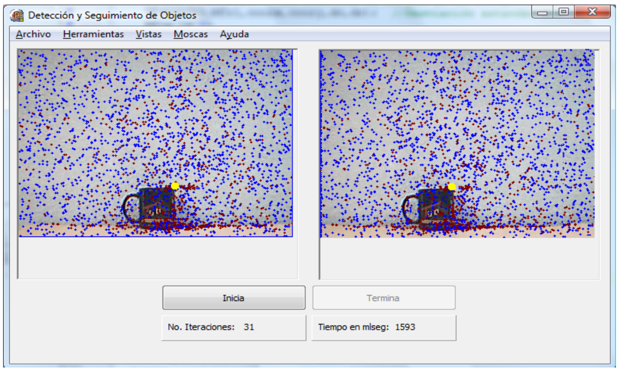
Fig. 7. Left and right images with initial population.


Detección y Seguimiento de Objetos
Archivo Herramientas Vistas Moscas Ayuda
Inicia
Termina

No. Iteraciones: 30

Tiempo en mlseg: 1530

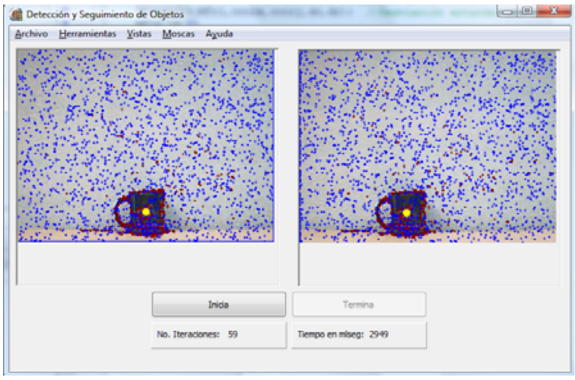
Fig. 8. Results after 30 executions using the first pair of images.


Detección y Seguimiento de Objetos
Archivo Herramientas Vistas Moscas Ayuda
Inicia
Termina

No. Iteraciones: 31

Tiempo en mlseg: 1593

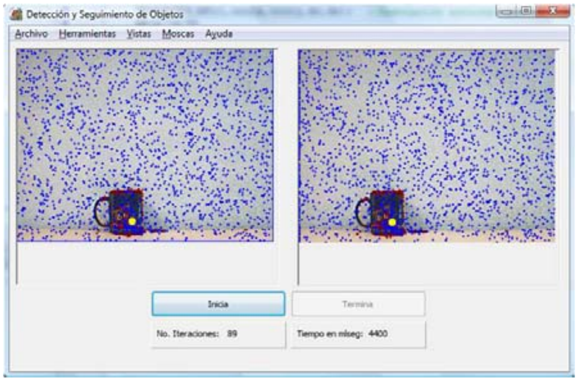
Fig. 9. The refresh of images, the flies are no longer posed on the object.


Detección y Seguimiento de Objetos
Archivo Herramientas Vistas Moscas Ayuda
Inicia
Termina

No. Iteraciones: 59

Tiempo en miseg: 2949

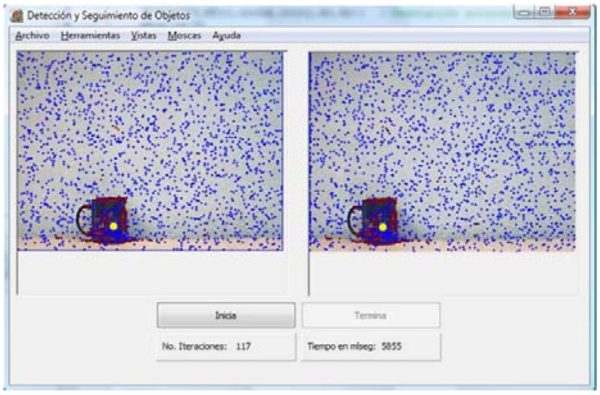
Fig. 10. Results after 59 executions using the second pair of images.


Detección y Seguimiento de Objetos
Archivo Herramientas Vistas Moscas Ayuda
Inicia
Termina

No. Iteraciones: 89

Tiempo en miseg: 4400

Fig. 11. Results after 89 executions using the third pair of images.


Detección y Seguimiento de Objetos
Archivo Herramientas Vistas Moscas Ayuda
Inicia
Termina

No. Iteraciones: 117

Tiempo en miseg: 5855

Fig. 12. The final result of the last pair of images after 117 executions.

8 Conclusion

We have presented a system for objects detection and tracking based on the fly algorithm and stereoscopy. We employ monocular vision that means only one camera is employed to capture the pair of stereo images. The camera displacement is carefully controlled. This system allows easy manipulation of all parameters involved in the process: the camera, the genetic algorithm parameters and the obtained centroid for the object detection and tracking. The system works very well and the object detection and tracking is obtained in real-time.

Finally, this system is being considered to be used in a mobile robot with monocular vision for navigation tasks or objects detection and tracking.

References

1. Álvarez, W.: Seguimiento de objetos para la reconstrucción de Ambientes en 3D. Universidad de Colombia (2007)
2. De Berg, M., Cheong, O., Van Kreveld, M., Overmars, M.: Computational Geometry, Algorithms and Applications. Heidelberg, Berlin, Springer, 3^{a}. Ed. (2008)
3. Blundell, B.G.: An Introduction to Computer Graphics and Creative 3-D Environments. Auckland, New Zealand, Springer (2008)
4. Boumaza, A.M. and Louchet, J.: Dynamic Files: Using Real-Time Parisian Evolution Robotics. In: EvoWorkshops (2001)
5. Chacón M.: Procesamiento Digital de Imágenes. México, Trillas (2007)
6. Gonzáles, R., Woods, R.: Tratamiento Digital de Imágenes. EUA, Addison Wesley (1992)
7. Hueso, D. l.: Visión por Computador. Fundamentos y Métodos. Prentice-Hall (2001)
8. Koza, J.: Genetic Programming II: Automatic Discovery of Reusable Programs. The MIT Press (1992)
9. Louchet, J., Guyon, M., Lesot, M., Boumaza, A.: Dynamic files: a new pattern recognition tool applied to stereo sequence processing. (2002)
10. Pajares, G., De la Cruz, J. M.: Visión por Computador: Imágenes Digitales y Aplicaciones. México, Alafaomega (2002)
11. Pauplin, O., Louchet, J., Lutton, E., Parent, M.: Obstacle Detection by Evolutionary Algorithm: the Fly Algorithm. Palmerston North, New Zealand (2004)

A New Optimization Method
Based on a Paradigm Inspired by Nature

Leslie Astudillo, Patricia Melin, and Oscar Castillo

Tijuana Institute of Technology,
Tijuana, Mexico
leslie.astudillo@suntroncorp.com, epmelin@hafsamx.org, ocastillo@hafsamx.org

Abstract. In this paper, we propose a new optimization method for soft computing problems, which is inspired on a nature paradigm: the reaction methods existing on chemistry, and the way the elements combine with each other to form compounds, in other words, quantum chemistry. This paper is the first approach for the proposed method, and it presents the background, main ideas, desired goals and preliminary results in optimization.

Keywords: Optimization, Chemical Reactions.

1 Introduction

Several works have proved the relevance of computing techniques to solve diverse kinds of problems, including forecasting, control and pattern recognition among others [1-3].

These techniques not only comply with their objective, but they also promote the creation of new ways to give solutions and improve the actual methods as well [4-6].

One of the main difficulties when designing the structure of a solution method is the tuning of the parameters; which are the key to the success of these applications. These parameters will vary depending on the complexity of the problem and the method used to find the solution; and in some cases, they stem from our own ability to conceptualize the problem itself, taking in account, the inputs of the sys-tem and the expected output values.

Due to these facts, several optimization strategies based on nature paradigms have arisen. From Ant Colony Optimization, to Particle Swarm Optimization among others, these strategies had emerged as an alternative way to solve problems [7-13].

For this work, we will be observing the process in which the different elements existing in nature are created, behave and interact with each other to form chemical compounds.

The structure of this paper is the following. Section 2 shows a brief description of the chemical method that inspired this investigation; section 3 describes the proposed method and first approach; section 4 shows the preliminary experiment results; in section 5 we describe the current and future work and section 6 shows some references.

© G. Sidorov et al. (Eds.)
Advances in Soft Computing Algorithms
Research in Computing Science 49, 2010, pp. 23-35

2 Chemical Paradigm

In order to have a better understanding of the process that we intend to model, we present some general definitions [14][15].

Chemistry is the study of matter and energy and the interaction between them, including the composition, the properties, the structure, the changes which it undergoes, and the laws governing those changes. A substance is a form of matter that has a defined composition and characteristic properties. There are two kinds of substances: elements and compounds.

An element is a substance that cannot be broken down into simpler substances by ordinary means. It is apparent from the wide variety of different materials in the world that there are a great many ways to combine elements.

Compounds are substances formed by two or more elements combined in definite proportions through a chemical reaction. There are millions of known compounds, and thousands of new ones are discovered or synthesized each year.

A chemical reaction is a change in which at least one substance changes its composition and its sets of properties; they are classified into 4 types.

Type 1: combination reactions: (B+C → BC).

A combination reaction is a reaction of two reactants to produce one product. The simplest combination reactions are the reactions of two elements to form a compound. After all, if two elements are treated with each other, they can either react or not.

Type 2: decomposition reactions: (BC → B+C).

The second type of simple reaction is decomposition. This reaction is also easy to recognize. Typically, only one reactant is given. A type of energy, such as heat or electricity, may also be indicated. The reactant usually decomposes to its elements, to an element and a simpler compound, or to two simpler compounds.

Binary compounds may yield two elements or an element and a simpler compound. Ternary (three-element) compounds may yield an element and a compound or two simpler compounds. These possibilities are shown in the Figure 1.

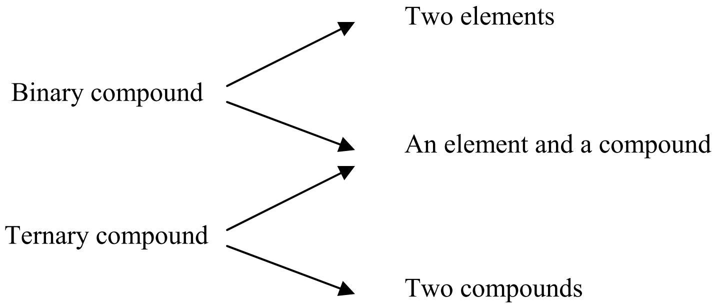


Fig. 1. Decomposition possibilities.

Type 3: substitution reactions: (C + AB → AC + B).

Elements have varying abilities to combine. Among the most reactive metals are the alkali metals and the alkaline earth metals. On the opposite end of the scale of reactivities, among the least active metals or the most stable metals are silver and gold, prized for their lack of reactivity. Reactive means the opposite of stable, but means the same as active.

When a free element reacts with a compound of different elements, the free element will replace one of the elements in the compound if the free element is more reactive than the element it replaces. In general, a free metal will replace the metal in the compound, or a free nonmetal will replace the nonmetal in the compound. A new compound and a new free element are produced.

Type 4: double-substitution reactions: (AB + CD → CB + AD).

Double-substitution or double-replacement reactions, also called double-decomposition reactions or metathesis reactions, involve two ionic compounds, most often in aqueous solution. In this type of reaction, the cations simply swap anions. The reaction proceeds if a solid or a covalent compound is formed from ions in solution. All gases at room temperature are covalent. Some reactions of ionic sol-ids plus ions in solution also occur. Otherwise, no reaction takes place.

Just as with replacement reactions, double-replacement reactions may or may not proceed. They need a driving force. In replacement reactions the driving force is reactivity; here it is insolubility or co-valence.

3 Modeling the Chemical Paradigm

Now that we have described the natural paradigm that we intent to mimic, the next step is to define the general structure of our optimization algorithm; which, initially will be developed in 5 phases: a combination algorithm, a decomposition algorithm, a substitution algorithm, a double-substitution algorithm and the final algorithm, which will be the combination of all the previous four.

The steps to consider in this optimization method will be as follows:

1. First, we need to generate an initial pool of elements/compounds.
2. Once we have the initial pool, we have to evaluate it.
3. Based on the previous evaluation, we will select some elements/compounds to “induce” a reaction.
4. Given the result of the reaction, we will evaluate the obtained elements/compounds.
5. Repeat the steps until the algorithm meets the criteria (de-sired result or maximum number of iterations is reached).

In order to start testing the phases of the algorithm, we will be applying these to the following (but not restricted to) functions: De Jong’s and Rosenberg’s functions [16], [12].

4 Preliminary Experimental Results

Figure 2 shows the De Jong’s first function also called the sphere model, which is continuous, convex, unimodal and is represented by the equation:

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \tag{1}$$

The domain is given by:

$$-\infty \leq x_i \leq \infty \tag{2}$$

And has a global minimum represented by:

$$f(x) = 0;\ x(i) = 0;\ i = 0:n. \tag{3}$$

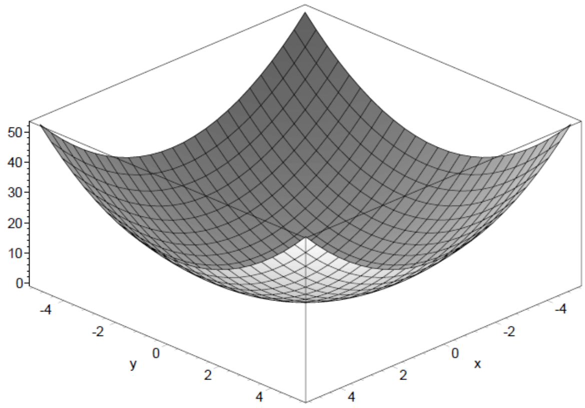


Fig. 2. De Jong´s First function in 2D.

The fist approach to solve this equation is given by applying a minimization algorithm based on the decomposition reactions.

The main idea of this particular algorithm is, given a random set of initial numbers, decompose each one into smaller numbers in a way that can be represented by a binary tree, where each ensuing node will be decomposed as well into smaller numbers, to lead the result into the minimum of the function.

To start from the simplest option, in these early experiments all decomposed elements are considered to have the same value, and they are given by:

$$\text{Decomposed\_Element(n)} = \text{Element/m} \tag{4}$$

Where *n* is the element index and m is the number of decomposed elements generated.

Because the resulting values are the same for each decomposed element, only one will be selected to be evaluated in the function.

Let’s consider an initial pool of 5 elements (randomly generated); each one will be decomposed in 3 sub elements throughout 10 iterations.

Table 1 shows the final and average values of the best and worst result reached by the algorithm throughout 30 experiments.

Table 1. Worst and best results throughout 30 experiments evaluating the first De Jong's Function.

Experiment number	Minimum Value	Average value	Comments
1	3.799e-14	1.656e-06	Best result
13	7.829e-09	0.341	Worst result

Figure 3 shows the minimized values trough the 10 iterations of experiment number 1, which reached the minimum value overall the 30 experiments.

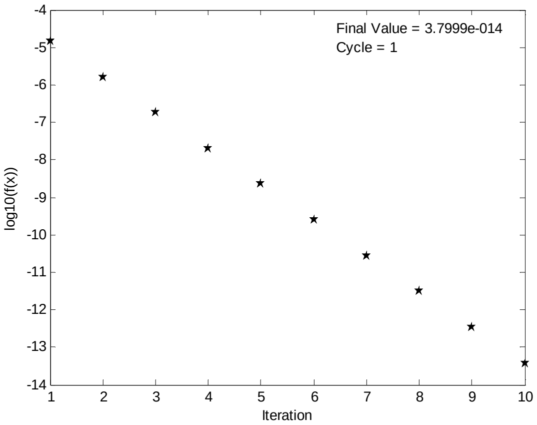


Fig. 3. Minimum value reached in experiment no. 1.

In Figure 4 we can see the behavior of the algorithm along the 30 experiments, where every experiment is represented by "Cycles" of 10 iterations each.

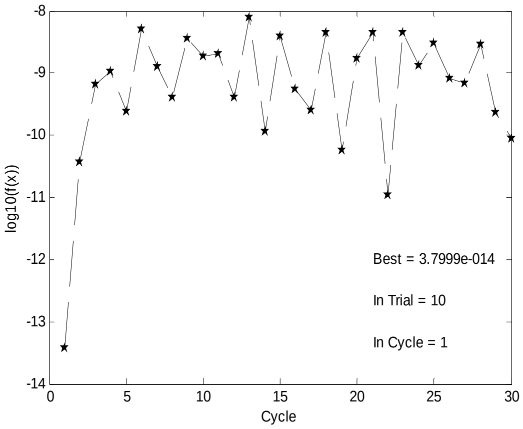


Fig. 4. Minimum values reached 30 experiments.

Table 2 shows the standard deviation calculated by iteration throughout 30 experiments.

Figure 5 shows the plot of the average and standard deviations calculated per iteration in 30 experiments.

Table 2. Standard deviation per trial in 30 experiments evaluating the first De Jong's Function.

Trial	Standard Deviation
1	0.769404299
2	0.085489367
3	0.009498819
4	0.001055424
5	0.000117269
6	1.30E-05
7	1.45E-06
8	1.61E-07
9	1.79E-08
10	1.99E-09

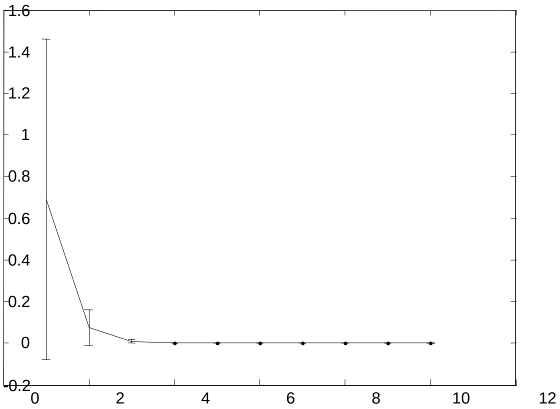


Fig. 5. Plot of the average and standard deviations per iteration in 30 experiments evaluating the De Jong's first function: *the sphere model.*

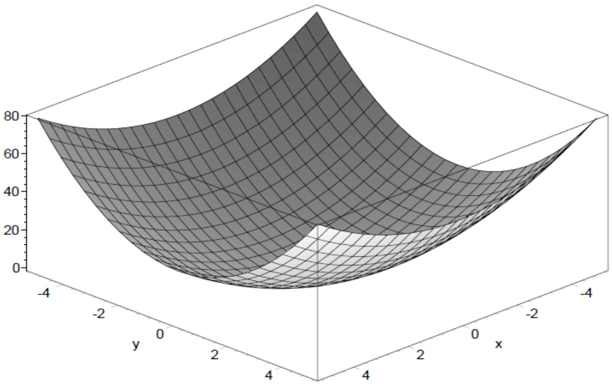


Fig. 6. The weighted sphere model in 2D.

Figure 6 shows the axis parallel hyper-ellipsoid, also known as the weighted sphere model. It a continuous, convex and unimodal function and is represented by the equation:

$$f_1(x) = \sum_{i=1}^{n} x * x_i^2 \tag{5}$$

The domain is given by:

$$-\infty \leq x_i \leq \infty \tag{6}$$

And has a global minimum represented by:

$$f(x) = 0; \;\; x(i) = 0; \;\; i = 0 : n. \tag{7}$$

The Table 3 shows the final and average values of the best and worst result reached by the algorithm throughout 30 experiments.

Table 3. Worst and best results throughout 30 experiments evaluating the first De Jong's Function.

Experiment number	Minimum Value	Average value	Comments
28	8.85E-17	7.00-05	Best result
23	7.91-13	0.62	Worst result

Figure 7 shows the minimized values trough the 10 iterations of experiment number 28, which reached the minimum value overall the 30 experiments.

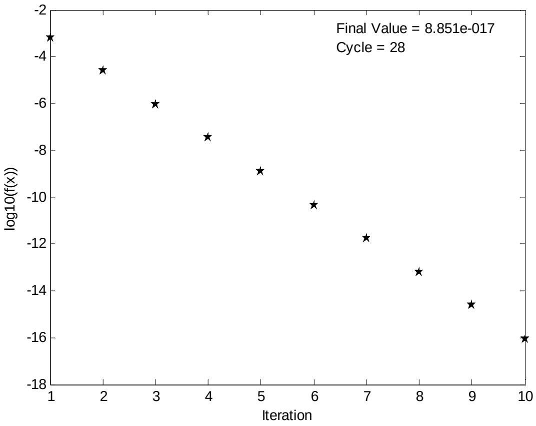


Fig. 7. Minimum value reached in experiment no. 28, evaluating the weighted sphere model.

In Figure 8 we can see the behavior of the algorithm along the 30 experiments, where every experiment is represented by "Cycles" of 10 iterations each.

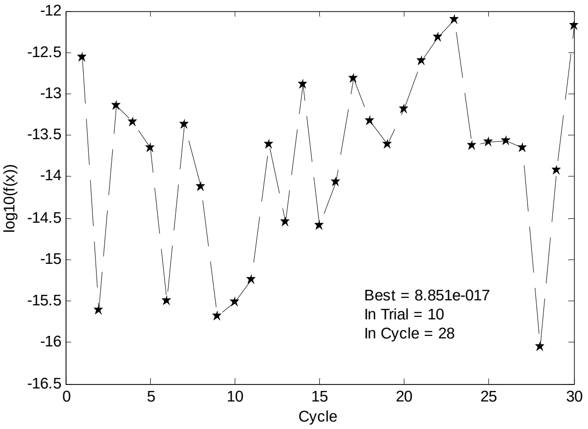


Fig. 8. Minimum values reached 30 experiments evaluating weighted sphere model.

Figure 9 shows the plot of the average and standard deviations calculated per iteration in 30 experiments, evaluating the weighted sphere model.

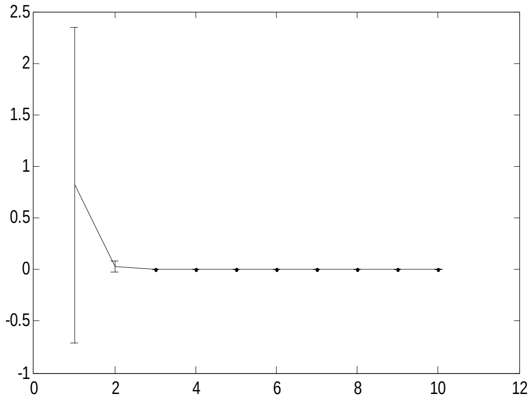


Fig. 9. Plot of the average and standard deviations per iteration in 30 experiments evaluating the weighted sphere model.

Table 4 shows the standard deviation calculated by iteration throughout 30 experiments.

Table 4. Standard deviation per trial in 30 experiments evaluating the weighted sphere model.

Trial	Standard Deviation
1	1.525
2	0.056
3	0.0020
4	7.75e-05

5	2.87e-06
6	1.06e-07
7	3.93e-09
8	1.45e-10
9	5.40e-12
10	2.00e-13

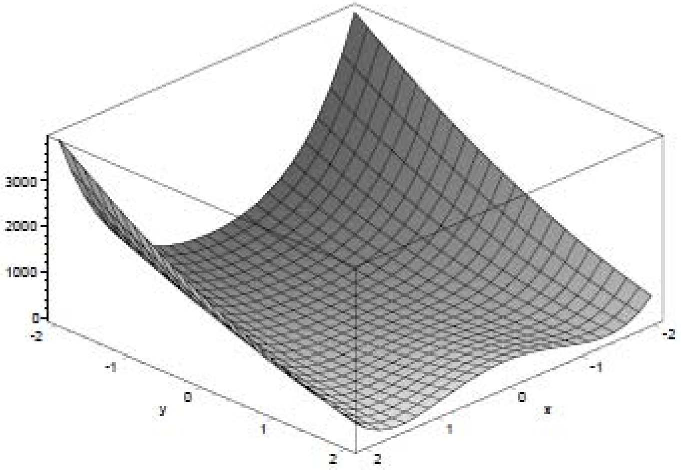


Fig. 10. The Rosembrock's Valley in 2D.

Figure 10 shows the Rosembrock's Valley function, also known as *banana function* or the *second function of De Jong.* The global optimum lies inside a long, narrow, parabolic shaped flat valley. It is represented by the equation:

$$f_1(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(1-x_i\right)^2\right] \qquad \textbf{(8)}$$

The test area is usually restricted to hypercube:

$$-2.048\le x_i\le 2.048 \qquad \textbf{(9)}$$

And has a global minimum represented by:

$$f(x)=0;\;\; x(i)=1;\;\; i=1:n. \qquad \textbf{(10)}$$

The Table 5 shows the final and average values of the best and worst result reached by the algorithm throughout 30 experiments.

Table 5. Worst and best results throughout 30 experiments evaluating the Rosembrock's Valley Function.

Experiment number	Minimum Value	Average Value	Comments

24	0.99966	1.053	Best final result
30	0.99997	1.013	Worst final result
17 (Iteration 2)	6.88e-06	1.23	Best result (Minimum value)

Figure 11 shows the minimized values trough the 10 iterations of experiment number 24, which reached the minimum final value overall the 30 experiments..

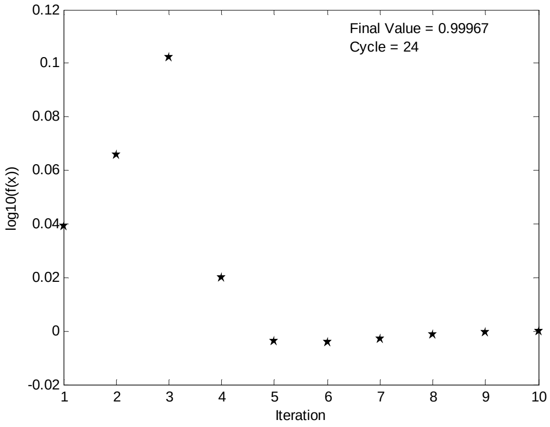


Fig. 11. Minimum value of experiment no. 24, evaluating the Rosembrock's Valley Function.

In Figure 12 we can see the behavior of the algorithm along the 30 experiments, where every experiment is represented by "Cycles" of 10 iterations each.

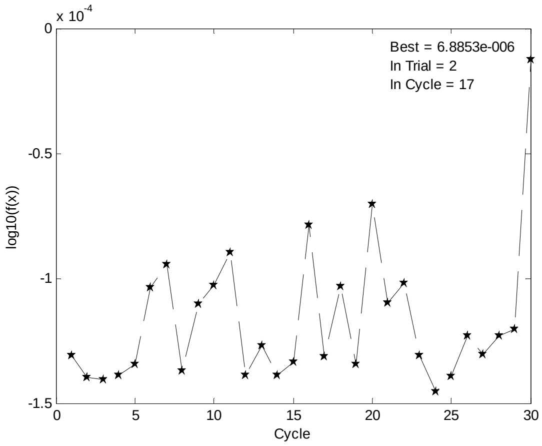


Fig. 12. Minimum values reached in 30 experiments evaluating Rosembrock's Valley Function.

As we can see in Table 5 and Figure 12, the last value reached in each experiment, may not be the "best value" found by the algorithm.

Figure 13 shows the behavior of trial with the minimum result through the 30 experiments.

In this experiment, the final value was 0.9997, but the minimum value (6.88e-06) was found in iteration no. 2.

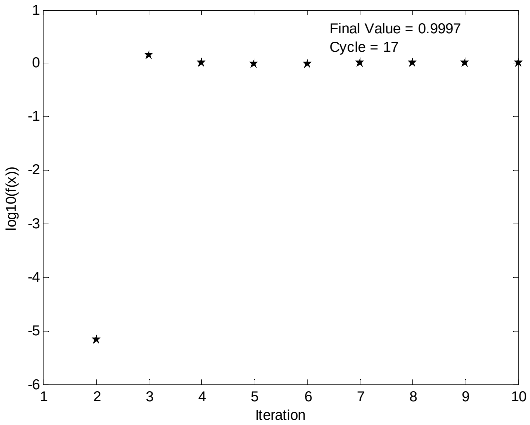


Fig. 13. Experiment 17, which had the minimum value in 30 experiments.

Table 6 shows the standard deviation calculated by iteration throughout 30 experiments.

Table 6. Standard deviation per trial in 30 experiments evaluating the Rosembrock's Valley Function.

Trial	Standard Deviation
1	13.443
2	0.796
3	0.205
4	0.024
5	0.004
6	0.0014
7	0.0012
8	0.00053
9	0.00018
10	6.44e-05

Figure 14 shows the plot of the average and standard deviations calculated per iteration in 30 experiments, evaluating the Rosembrock's Valley Function.

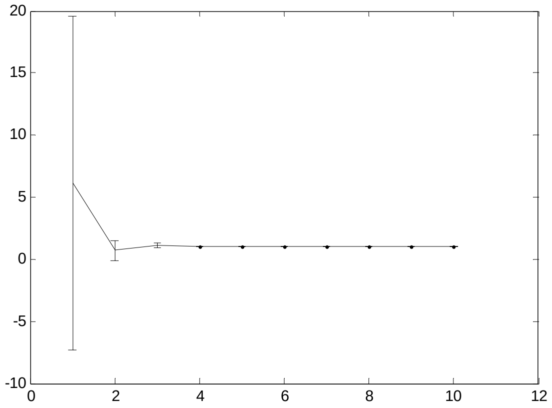


Fig. 14. Plot of the average and standard deviations per iteration in 30 experiments evaluating the Rosembrock's Valley Function.

5 Conclusions

In this paper, we introduced the first stage of a new optimization method that tries to mimic the chemical reactions. The *Decomposition Reaction Method* was applied in 3 benchmark functions to evaluate the first development phase of the optimization algorithm. This *Decomposition Reaction Method* by itself finds or guides the result to a certain minimum value, but, due the nature of some functions, it is necessary to introduce the second phase of this optimization method: *The Combination Reactions Method,* which will be able to guide the algorithm to find an optimum value when it is not necessarily the "smallest" one. At the time, more functions are being evaluated to pursue the tuning of the algorithm itself.

Acknowledgment. The authors would like to thanks CONACYT and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

1. Hidalgo, D., Melin, P., Licea, G.: Optimization of Modular Neural Networks with Interval Type-2 Fuzzy Logic Integration Using an Evolutionary Method with Application to Multimodal Biome-try. Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition. pp. 111-121 (2009)
2. Astudillo, L., Castillo, O., Aguilar, L.: Hybrid Control for an Autonomous Wheeled Mobile Robot Under Perturbed Torques. In: Foundations of Fuzzy Logic and Soft Computing. IFSA Proceedings, pp. 594-603 (2007)

3. Melin, P., Mancilla. A., Lopez, M., Solano, D., Soto, M., Castillo, O.: Pattern Recognition for Industrial Security Using the Fuzzy Sugeno Integral and Modular Neural Networks. Advances in Soft Computing. Vol. 39. No. 1., 105-114 (2007)
4. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks. Piscataway, NJ., pp. 1942-1948 (1995)
5. Valdez, F., Melin, P.: Parallel Evolutionary Computing using a cluster for Mathematical Function Optimization. In: Nafips, San Diego CA, USA. pp. 598-602. (2007)
6. Fogel, D.B.: An introduction to simulated evolutionary optimization. IEEE transactions on neural networks. Vol. 5, No. 1., 3-14 (1994)
7. Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms: Concepts and Designs. Springer Verlag (1999)
8. Eberhart, R. C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science. Nagoya, Japan. pp. 39-43 (1995)
9. Goldberg, D.: Genetic Algorithms. Addison Wesley (1988)
10. Angeline, P. J.: Using Selection to Improve Particle Swarm Optimization. In: Proceedings of the World Congress on Computational Intelligence. Anchorage, Alaska. IEEE, pp. 84-89 (1998)
11. Montiel, O., Castillo, O., Melin, P., Rodriguez, A., Sepulveda, R.: Human evolutionary model: A new approach to optimization. Inf. Sci. 177(10), 2075-2098 (2007)
12. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. Second Edition. Wiley Interscience (2004)
13. Rotar, C.: A New Evolutionary Algorithm for Multiobjective Optimization Based on the Endocrine System. In: Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics – ICTAMI, Alba Iulia (2003)
14. Chang, R.: General Chemistry. 5th edition. McGraw-Hill (2004)
15. Goldberg, D.: Schaum's Outline of Beginning Chemistry. 3rd edition. Schaum's Outline Series. McGraw-Hill (2009)
16. GEATbx: Example Functions (single and multi-objective functions), http://www.geatbx.com/docu/fcnindex-01.html#P89_3085

Specialized Algorithms

An Insect Flight as Non-Holonomic System

Abraham Rojano[1], Raquel Salazar[1], Irineo López[1], and Fernando Rojano[2]

[1] Universidad Autónoma Chapingo Km 38.5 Carretera México-Texcoco Chapingo, Edo. de México, C.P 56230,
Mexico
[2] Faculty of Biosystem Engineering,
University of Arizona, Tucson AZ, 85712
USA
abrojano@hotmail.com

Abstract. Nowadays, with currently available computational resources, it is possible to tackle complicated problems of not linear type. The specific case of the flight of an insect is not the exception, and although the problem is extremely daily, it can be an excellent testing to show the involved spatial and time trajectories not only as an elegant challenge but also serves to illustrate the process of abstraction by assuming appropriate hypothesis and using elements of the classic mechanics, in order to yield into the solution process. Based on Newton-Euler equations with four primary variables like position of the insect, horizontal and vertical velocity, angle, and angular velocity the simulation of longitudinal flight is carried out. These simulations consider the effect of different velocities of the air and also different angles of attack. Finally, for the nonlinear time invariant system there is a 3D plot showing patterns in how the position of the insect is changing with reference to the other three variables considered in the system.

Keywords: Computational, Newton-Euler, Velocity, Pattern.

1 Introduction

Nonholonomic systems are studied from the Euler times in the 18th century, and they are mainly concerned with frictionless problems corresponding to motion of rigid bodies. Euler, Lagrange and Hamilton point of views are commonly linked by means of Poisson brackets, however by using special constraints in position or velocity requires generalizations preserving the energy system, it requires the combination of experiments, mathematics, and computation, in order to simulate different scenarios. Even though, conservation laws in biology are difficult to identify, it is possible to assume that insects are somehow symmetrical as well as its body is made of rigid segments subject to assumptions of potential flow surrounding the insect. Similarity with the studies of aircrafts navigation like airplanes or helicopters, as well as its numerously industrial applications, nonholonomic systems enhance and motivate exhaustive simulations in order to control by determining the ranges of stability. Furthermore, adapting aircraft stability to the flight of insects should be made by

© G. Sidorov et al. (Eds.)
Advances in Soft Computing Algorithms
Research in Computing Science 49, 2010, pp. 39-46

means of forces produced with the wings flapping. This implies that insects follow a particular dynamics beyond the simple geometry and kinematics, so this work involves a living system with wings subjected to the total force and/or moment produced [1].

Although, any insect follows similar flights or patterns, it is easily and quickly observed that size, shape, and weight are sufficiently different to be impossible achieve a realistic representation and general solution. Under these circumstances, we are starting to explore possibilities to simulate a flight of a particular locust [2]. In the first part, it is developed a methodology to log a set of data from experiments which are going to be used in a set of equations to reproduce a flight under a linearized model [1]. This set of data includes some properties like size, weight, moment of inertia, and forces produced by flapping.

1.1 Main Characteristics of the Locust

As the model is determined by physical properties of the locust, then it is important to cite some of them. For example, to have an idea the wing beat frequencies of the largest butterflies is around (c. 40 rad/s in the 1 × 10−3 kg birdwing Troides rhadamantus) which is comparable to the rotor frequencies of the smallest helicopters (44 rad/s in the 2×103 kg Eurocopter Bo1054) [3].

The desert locust is a robust, migratory, four-winged insect of 0.1m span, optimized for endurance, rather than maneuverability, and with a range of several kilometers at its cruise speed of 4 m/s. Although locusts do not vary in overall body plan, different individuals vary markedly in size. The wings beat at c. 20 Hz, with the hindwing leading the forewing by π/6 rad. The hindwing typically sweeps 110° through every half-stroke, while the forewing sweeps through 70°. The hindwing comprises a collapsible fan consisting of a flexible membrane with supporting ribs radiating from the root. Each wing is moved by 10 muscles, and with no separate control surfaces, flight control is made by changes in the wing kinematics. Slower flight control is made by moving the legs and abdomen, which act as a rudder and also shift the center of mass. Locusts are equipped with a range of sensors known to be used in correctional flight control. The antennae sense airspeed, analogous to a Pitot tube. Wind-sensitive hairs on the head sense aerodynamic incidence. Each wing has at its base strain receptors involved in flight control that may measure total wing loading, but perhaps more likely monitor wing kinematics: they have no direct or functional analogy in conventional aircraft [3].

From observations of biologists it is pointed out that compound eyes works as horizon detectors measuring bank and pitch angle, and as optic flow detectors measuring heading and perhaps pitch, roll and yaw rate. Additionally, locusts are not known to have any sense of gravity in flight. Although vision is certainly important in insect flight control, it cannot give absolute measurements of bank and pitch angle unless referenced to gravity, and is ambiguous in respect of certain aspects of self-motion [3].

1.2 Newton-Euler Equations

To simulate the flight, this framework is going to be based on the set of equations of Newton-Euler obtained from the general principles of lineal and angular momentum conservation, respectively.

$$\begin{aligned}
\dot{u} &= -wq + \frac{X}{m} - g\sin\theta \\
\dot{w} &= uq + \frac{Z}{m} + g\cos\theta \\
\dot{q} &= \frac{M}{Iyy} \\
\dot{\theta} &= q
\end{aligned} \tag{1}$$

From this set of equations is important to say that simulation is going to consider a 2D simulation flight, longitudinal with respect to the insect. Formally, u and w are the velocity with respect to the horizontal and vertical axis z, correspondingly, q means the angular velocity with respect to the center of mass and θ is the angle with respect to the horizontal axis, as shown below in Fig1.

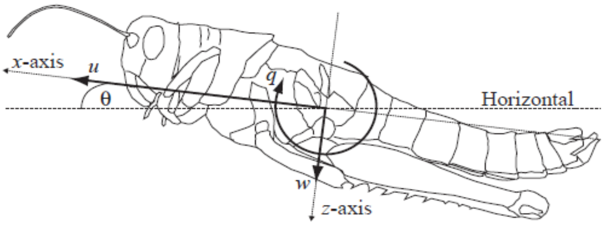


Fig. 1. Diagram of a two dimensional insect section.

2 Objectives

– To identify what are the most important factors affecting stability of insect flight.
– To develop a simulation program for nonlinear conditions identifying the central core data.

3 Methodology

There were several concerns regarding to data obtained from experiments. Basically, attending needs of the Newton-Euler equations some values as the weight, moment of inertia and forces should be as accurate as possible. Considering three types of locusts some data were considered in the simulation and that information can be summarized in Table 1 [1]. Although, in Table 1 considers three type of locust, considering simulations for the locust type “R”, “B”, and “G”.

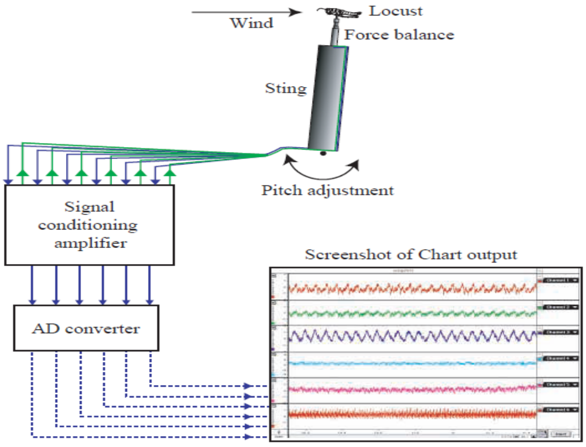


Fig. 2. Experimental setup for recording forces from a locust.

3.1 Logging forces

Logging forces are important since they are the reference in the simulation, for analyzing stability and forces. Essentially, a locust was attached to the top of a needle. That locust was inside of a wind tunnel (see Fig 2) and the forces were recorded at different wind speeds and also at different position of the insect (angle with respect to the horizontal axis). Forces were recorded in an interval from 2 to 5.5 m/s and also at angles from 0° to 14°; these ranges were given from observations when the locust is found in a free flight [1].

4 Results

Once forces were recorded, it was needed to preprocess data. This preprocess implies that was done a simplification of forces variability to one representative flapping. This means that there is an assumption that flapping is uniform during the flight and one period of flapping is going to be exactly equal to the next one; which it does not occur in a real flight but it reduces the set of data significantly. Then, data obtained from experiments, where variability of forces for one cycle of flapping (X correspond to the horizontal direction) at different angles of the locust (from 0° to 14°).

Data recorded include thousands of points difficult to handle, to overcome this situation was used standard Fourier series in order to reduce into a few coefficients corresponding to eight harmonics.

Forces in the horizontal and vertical directions were reproduced at different angles and moments for various wind speeds and appropriate references and right connections. With this new set of tools was looked for the quasi-static equilibrium in order to start simulations for quasi-static conditions with Runge-Kutta methods provided by Matlab toolbox as ode45 function.

One way to analyze performance of the model is considering a small perturbation in one value of the initial conditions in a linear time periodic model (LTP). For instance, horizontal velocity (u) is perturbed by a small value of 0.002 m/s during 0.4 s corresponding to 8 cycles of flapping) were it was found that among three types of locust follow the same pattern. These simulations consider average of the forces during one cycle of flapping, even more nonlinear time periodic (NLTP) models include force variations. NLTP model considers forces in Fourier series fitted until the eight order of harmonics. This covers variability of either forces or moments during one cycle of flapping, with initial conditions equal to the quasi-static equilibrium. Table 1 provides information regarding the physical and geometric characteristics of three type of locust, considering simulations for the locust type “R”, “B”, and “G”.

Table 1. Reference values for three types of locust.

Locust	Reference body mass (g)	Reference body length (mm)	Moment of inertia Iyy (10-9 kg/m2)
“R”	1.8490	46.0	232.8
“G”	1.4357	40.5	140.3
“B”	1.8610	46.0	236.0

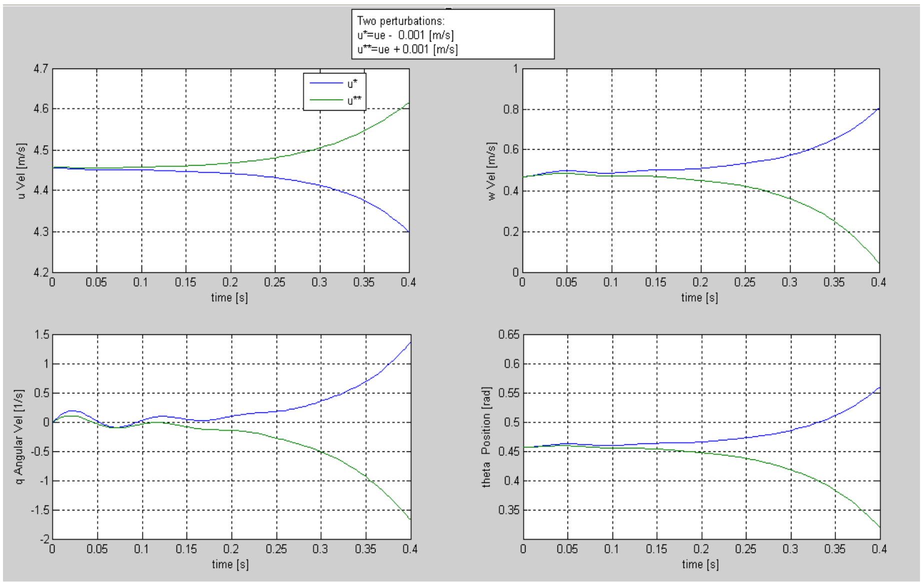
Table 1 provides information regarding the typical angles and velocities used for three locusts.

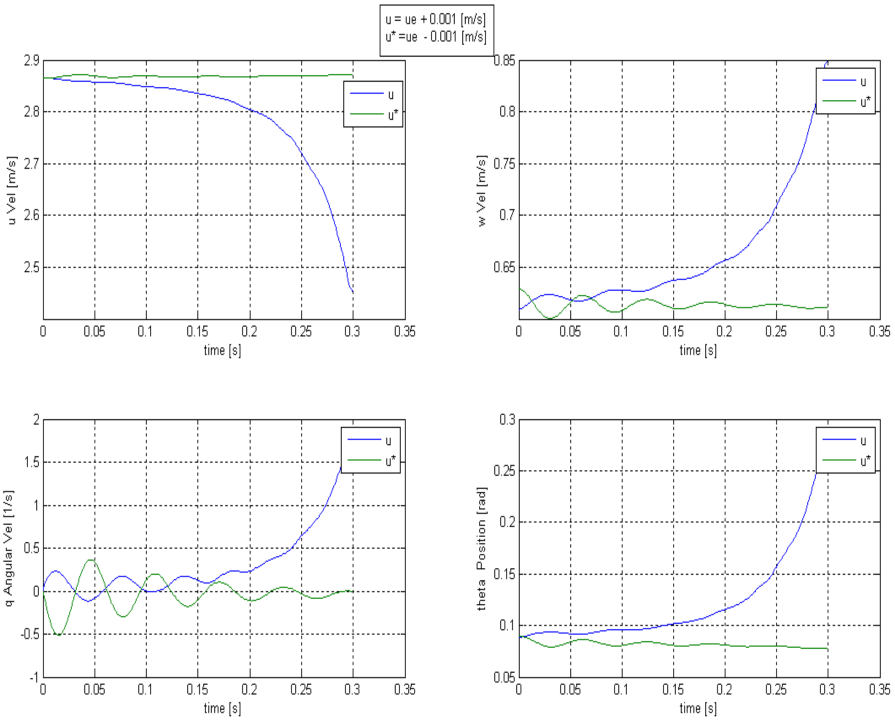
Table 2. Conditions for quasi-static equilibrium of each locust.

Locust	θ(deg)	α(deg)	U(m/s)
“R”	26	6	4.48
“G”	5	12	2.93
“B”	23	9	4.79

The simulations were made by perturbing the horizontal velocity, with initial time modified by 0.025s as equivalent to a half cycle of flapping. This perturbation accomplishes the idea of how much would change a possible trajectory of the flight because starting of flapping with open wings instead of closed wings (when initial time is equal to zero).

The simulated final path is completely different for locust type “R” because of different initial times. In the same way in Figure 12 there is a similar result. However, in Figure 11 which corresponds to locust type “G”, there is a parallel path for the four variables. This situation may tell that this type of locust might be stable.


Two perturbations:
u*=ue - 0.001 [m/s]
u**=ue + 0.001 [m/s]
u*
u**
u Vel [m/s]
4.7
4.6
4.5
4.4
4.3
4.2
w Vel [m/s]
1
0.8
0.6
0.4
0.2
0
q Angular Vel [1/s]
1.5
1
0.5
0
-0.5
-1
-1.5
-2
theta Position [rad]
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4
time [s]


u = ue + 0.001 [m/s]
u* =ue - 0.001 [m/s]
u
u*
u Vel [m/s]
2.9
2.8
2.7
2.6
2.5
w Vel [m/s]
0.85
0.8
0.75
0.7
0.65
q Angular Vel [1/s]
2
1.5
1
0.5
0
-0.5
-1
theta Position [rad]
0.3
0.25
0.2
0.15
0.1
0.05
0 0.05 0.1 0.15 0.2 0.25 0.3 0.35
time [s]

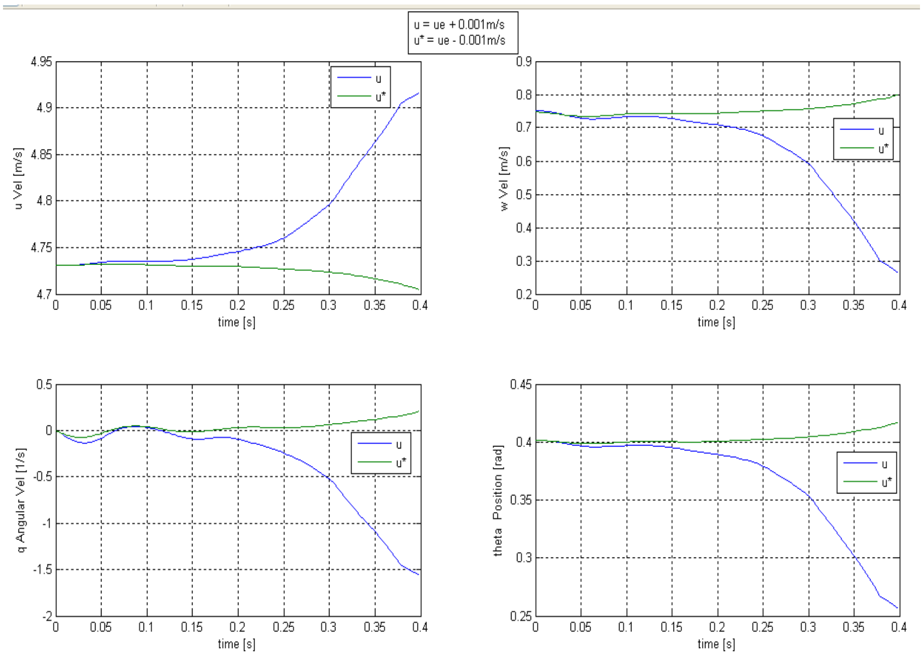


Fig. 3. Simulation for a Non linear time invariant model with perturbation in the u-velocity for locust types "R", "G", and "B", from top to below, respectively.

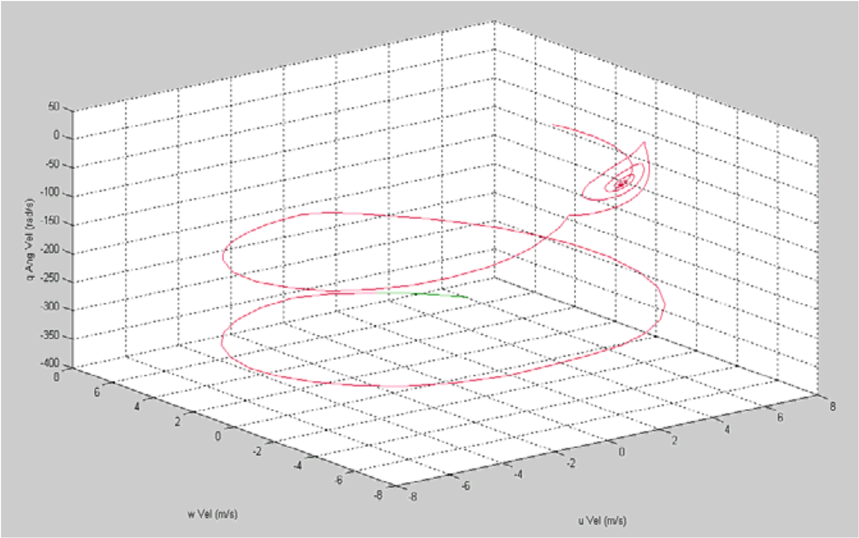


Fig. 4. Phase portrait for θ for the non linear time invariant model for locust type "R".

Finally, it is showed in Fig. 4 the variability of θ with respect to the other three variables (u, w and q). Based on the NLTI model; There are at least two points where θ changes abruptly when simulation time correspond to 0.6s. These rapid changes in the path of θ implies that once the insect falls in this point then the insect might have change rapidly the path. This situation does not happen in reality, due to instability in the simulation with a set of small spirals which also might be difficult to see in a free flight.

5 Conclusions

This is possible, since several factors were limiting it; for instance, forces were not accurate because of the instruments used, the insect can have different reactions under this "altered conditions" to simulate the flight. Additionally, some assumptions were made as the gravity, weight and moment of inertia are constant which is not true and it is easily noticed in the moment of inertia because of the flapping.

Also, when the experiments were performed, there was a recording of forces in an open loop. Under this situation, it is assumed that the insect does not control the flight, which is not true. Thus forces to manipulate the flight might already be included somehow in the set of data given in the experiment. Additionally, exploring possibilities to control the flight opens the chance to improve methodology to obtain accurate values from experiments and to estimate forces from the set of Newton Euler equations, combined with Extended Kalmann Filters reported by [4].

Even though the model is still weak to show a realistic simulation of an insect flight, methodology to obtain data is strongly improved and computationally algorithms are nowadays very helpful.

References

1. Taylor, G.K., Thomas, A.L.R.: Dynamic flight stability in the desert locust *Schistocerca gregaria*. Journal of Experimental Biology, 206, 2803-2829 (2003)
2. Taylor, G.K., Zbikowski, R.: Nonlinear time-periodic models of the longitudinal flight dynamics of the desert *Schistocerca gregaria*. Journal of the Royal Society Interface 2, 197-221 (2005)
3. Taylor, G.K., Bomphrey, R.J., Hoen, J.: Insect flight dynamics and control. In: Aerospace Science Meeting and Exhibit, Reno Nevada (2006)
4. Garrat, M.: Biologically Inspired Vision and Control for an Autonomous Flying Vehicle. Ph D. Dissertation (2007)

A Search Method using Jumping Behavior

Paola Neri-Ortiz[1] and Rubén de la Mora-Basáñez[2]

[1]Soluciones Informáticas y Diseño SID7.0
Francisco del Paso y Troncoso 19 D -7, Col. Jardín Balbuena, 15900, D.F.,
Mexico
[2]Facultad de Física e Inteligencia Artificial,
Sebastián Camacho No.5 Col. Centro, 91000, Xalapa, Ver.,
Mexico
paolaneriortiz@yahoo.com, carlosvub@hotmail.com

Abstract. Jumping behavior consists of an interleaving sequence of conducts of living organisms, such as moving or sniffing. Jumping behavior is used in many activities vital to subsistence, such as: food foraging, search for prey, avoid predators, etc. There are several interpretations for this behavior from the ethological viewpoint. Nevertheless, there is no clear biological model that provides comprehension of this conduct. We have developed a kinetic model to simulate this behavior as one that maximizes the probability to find an object in an open infinite space. The advantage of this model lies in its ability to find a local max number of goals in shorter time. Jumping algorithm uses few resources, as compared to ant algorithm which needs several agents to finding goals. Using the kinetic model, this jumping behavior (proper of biological entities) will be used as a search method for patterns in a two dimensional space, carried out by synthetic agents. Kinetic factors to consider are: moving time, local searching time, size of step, and perception capabilities. The purpose of this work is to study the behavior of the model of the jumping conduct, considered as a search method of patterns in closed 2-D spaces, through an incremental methodology. For this purpose a simulator was designed, in it, an agent looks for 2-D patterns using the leaping kinetic model. Results obtained in this work can be used in Robotics to approximate the region that a group of robots can sweep; or as an exploration method of non-unstructured spaces, such as it occurs in internet search and images search.

Keywords: Jumping Behavior, Search Methods, 2-D Pattern.

1 Introduction

1.1 Jumping Behavior

The biological phenomenon of intermittent locomotion describes the manner in that animals move, during the execution of their daily chores, such as feeding, having intercourse, prey searching, etc. [1]. An organism that shows intermittent locomotion typically alternates movement with phases of rest. Such behavior is seen, for instance, in squirrels exploring a park, birds that glide looking for prey, dogs sniffing for buried bones, etc.

We also would like to introduce some terms:

Time test: Cycles number than an agent executes the search of goals in a selected space.

Advance: Space agent travels in the same direction, measured in pixels.

Sniffing: Space in which the agent performs a local search, measured in pixels.

Size of the pattern: Pattern width and height measured in pixels.

Density pattern: Pattern density measured in pixels.

Space dimension: Width and height of the space, measured in pixels.

Space density: Space density measured in pixels.

1.2 Kinetic Model

Although there is no clear biological model that explains such behavior, a model that describes it exists [2]. The kinetic model of jumping behavior consists in an alternating sequence of advance (move) behavior and sniffing (or search) behavior, which maximizes the probability of finding objects in an open infinite space.

In the advance (move) phase, the organism moves in some random direction during a random time with average of τ_{move}. In the sniffing phase (local search), the organism pauses its movement in order to perceive or smell for food or prey, or in order to explore the unknown space surrounding it. The time spent on this activity is a variable random with a mean value $\tau_{sniffing}$. Next we explain the process through an algorithm consisting of a search method for 2-D patterns.

1.3 Jumping Behavior as Search Method

We apply the jumping behavior metaphor to a search algorithm that an agent uses in order to find 2-D patterns. Each life cycle of the agent consists of moving periods alternating with searching or “sniffing” periods. The agent moves through the search space using random values that are generated, corresponding to moving times and searching times (local search or sniffing). In the advance stage, the agent moves in a given direction, randomly chosen, during a random time generated with a mean τ_1 and a standard deviation σ_1. In the sniffing stage, the agent uses a random variable with mean τ_2 and standard deviation σ_2, to delimit a window where the local search for the sought pattern is performed.

The agent uses moving times to explore the search space, and sniffing times to make a local search. According to the used metaphor, the algorithm looks for a relation between moving frequency and sniff frequency that maximizes the probability of finding goals, at the same time, uses the lowest amount of resources, where moving and sniffing are supposed as restrictions.

Another algorithm based on the animal behavior is the ant algorithm. The ant algorithm is a heuristic technique used to solve problems where the search space is not delimited and non-structured [5]. In nature the ants go randomly in searching of food and when returning to the colony with the reward they release pheromones to guide the others by a successful way. With time, some ways will be but clear than others. This type search requires that several ants return to the departure point before visualizing a route clearly. Therefore the time of route for each ant is double the time it spends in finding the objective. In the algorithm of jump described in this work, the agent gets the objective and it marks it of a different color, this avoids that it finds the same objective twice (and wastes time).

Below we explain the theory of agents used as computational model.

2 Search Algorithm with Jumping Behavior

2.1 Behavior of the Agent during the Search

The agent used in this work searches 2-D patterns using the algorithm that is explained below, in an isolated form during the test phase and in collaborative fashion during the second stage. This agent that searches goals exhibits a jumping behavior, and alternates two conducts; a moving or advance conduct, and a sniffing or local search conduct.

The algorithm is as follows:

1) The agent starts at an initial position.
2) It selects a direction α, a distance factor T_{move} and size factor T_{search}, all random (mean values in τ_{move} and τ_{search}, standard deviation at 20%). The mean values are selected by user.
3) It moves in the chosen direction with a step of length L_{move} (selected by user), with

$$x = T_{move} \cos\alpha \qquad \text{y} \qquad y = T_{move} \, L_{move} \operatorname{sen}\alpha$$

4) The agent uses T_{search} (size factor) and L_{search} (selected by user) to performs a local search in a window of size

$$x = y = \sqrt{T_{search} \, L_{search} \, \pi}$$

If the goal is found during this stage, a counter is incremented.

5) The iterations number is reduced, if this is larger than cero go to step 2. Otherwise, end the program.

Below some steps of the algorithm are explained in detail:

1) At the start, the agent is positioned in the upper left corner 0, 0.
2) Then, an angle α is selected at random, used to change the direction of advance of the agent. A random value T_{move} is generated (with mean in

τ_{move} and standard deviation at 20%), the agent will use it like a distance factor.

3) During the advance behavior, the agent uses T_{move} and a length of the advance step L_{move} in order to move in the direction α, in this form:

$x = T_{move} \cos\alpha$ y $y = T_{move}\, L_{move} \operatorname{sen}\alpha$

4) During the sniffing behavior, the agent uses a random value T_{search} with average τ_{search} and standard deviation at 20%, and a step length L_{search} to search the goal in local form in a window of size.

$$x = y = \sqrt{T_{search}\, L_{search}\, \pi}$$

This relation was obtained by adjusting the local search area, as the area of a square that contains a cycle of radius L/2, as

$$Surface = L^2 = T_{search}\, L_{search}\, \pi$$

Where T_{search} is the sniffing (or local search) time, L_{search} is the step length and π is the constant relation that exists between the perimeter and the diameter of a circle.

2.2 Experimental Setting

To carry out the tests of the behavior of a search agent looking for 2-D patterns with the described behavior, an environment with variable density and size is required.

The environment was implemented in Java, as an interface that the search agent uses to show its behavior in graphical form. The interface has the ability to create a space and a 2-D pattern of variable size and density, and to insert that pattern in the space with another variable density. The system interprets the space and the pattern as a matrix of dots and stores it in a text file that can be reused in further executions. Fig. 1 shows the interface of the system.

The text boxes “Space Dimension and Space Density” modify the size in pixels and the space density. The values of pattern dimension and pattern density modify the size in pixels and the density of the pattern.

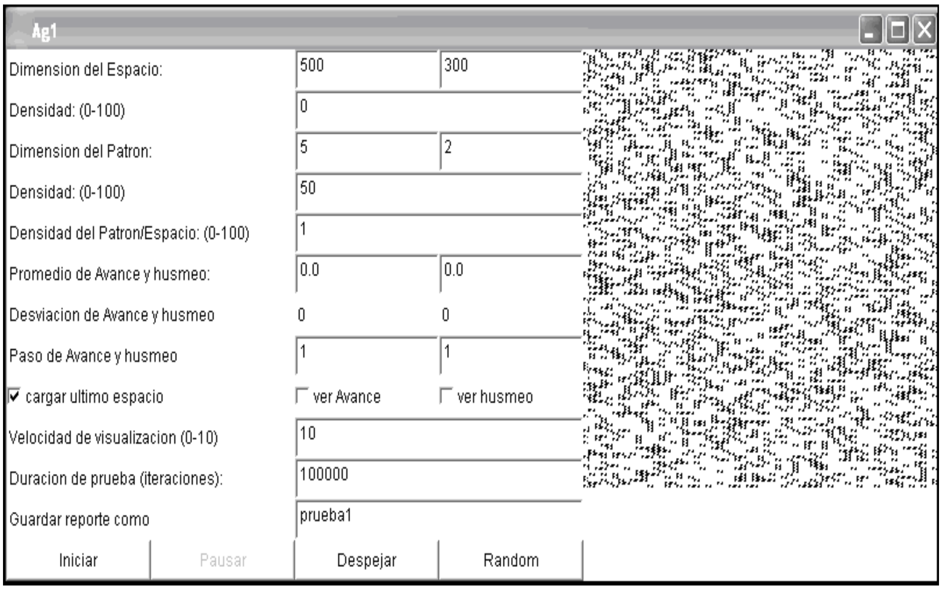
Pattern density value, establishes the probability with which the pattern will be inserted into the space.

The average values of moving and searching variables are used to generate random values of frequency with means in the selected points.

The number of iterations that the search agent will execute through the environment can be manipulated with the value “Cycles Number” once the test finishes, a report with the average of the moving and searching values, and the number of goals obtained, will be added to the file specified in “Save report as…”.

The “pause” button stops the execution of the search agent.

The “clear” button cleans the path of the agent, for easier identification of the patterns already found.



Dimension del Espacio:	500	300
Densidad: (0-100)	0	
Dimension del Patron:	5	2
Densidad: (0-100)	50	
Densidad del Patron/Espacio: (0-100)	1	
Promedio de Avance y husmeo:	0.0	0.0
Desviacion de Avance y husmeo	0	0
Paso de Avance y husmeo	1	1
☑ cargar ultimo espacio	□ ver Avance	□ ver husmeo
Velocidad de visualizacion (0-10)	10	
Duracion de prueba (iteraciones):	100000	
Guardar reporte como	prueba1	

Fig. 1. Experimental environment.

The "random" button generates a pair of random numbers between 0 and 100, used as the average moving frequency and searching frequency. The standard deviation for both frequencies has been adjusted at 20%.

The "Init" button generates the space according to the established values, and draws it. If the verification box labeled as "load last space" is selected, the system will load the text files that contain the last space and pattern used previously, instead of generating a new one. Once the space has been generated or loaded, a sequence of 100 random numbers is created, having as mean value the values of moving and searching and standard deviation of 20%. These values will be used to simulate the behaviors of moving and sniffing as explained now.

Fig. 2 is an example of the agent's behavior during the execution of the program. The space with inserted patterns is shown as dots in the background. The path of the agent is shown as lines and squares. The agent marks the patterns that have been already found as dark zones. The straight lines represent the advance stages, while the squares represent the periods of sniffing or local search.

Note that with respect to the used metaphor, long straight lines show that the agent requires more resources, for instance a larger amount of energy, or, if it were a robot, then it would require a large energy source. Analogous, large searching times mean local searches in large spaces, therefore longer times.

On the other hand, smaller advance times, will cause the agent to require much time to cover the total space. Small searching times will provoke that the agent does not take advantage of the energy that employed in moving in order to search a large amount of goals.

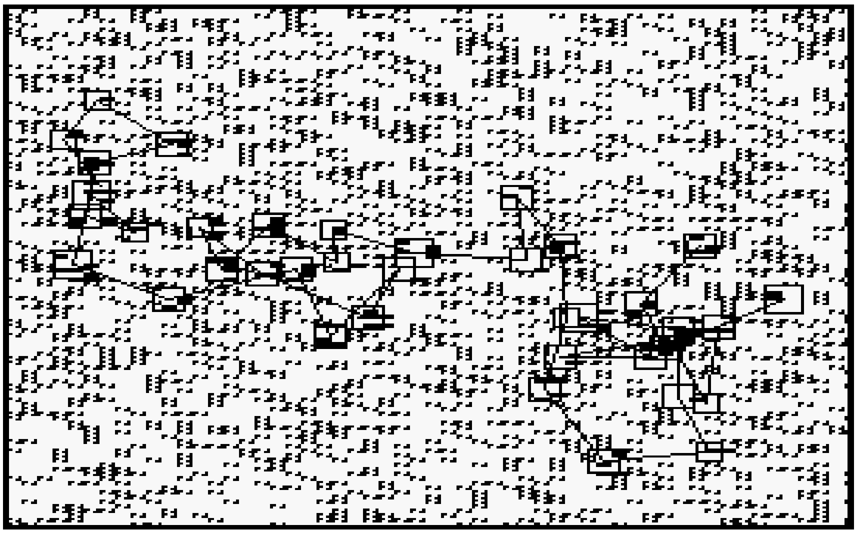


Fig. 2. Execution of the search agent.

3 Performance Test Results

So we were looking for a pair of values of moving frequency and searching frequency that uses smaller resources, and at the same time, maximizes the probability of finding goals.

The test consisted on making run to an agent in a space with 450 different combinations from moving and searching times. The purpose is to find those combinations that generate a large amount of goals.

3.1 Finding Goals in a Space of 500x300 pixels

- It was designed a pattern of 5x2pixels and inserted in a space of 500x300pixels with a density to 1%.
- In the previous space, it was executed an agent with a combination of moving and searching values chosen.
- The moving and searching values are in a range from 10 to 150 units for the advance and of 10 to 300 units for the sniff. The combinations were formed with 10 distance units, in the way (10,10), (10,20), (10,30), etc.
- For each combination of values (move, search) that form the sample, the goals were averaged found by an agent in 10 executions.
- Each execution had a duration of 100 000 iterations.
- Then a chart was built with the number average of goals found by the agent in each one of the combinations (move, search).

- In the previous chart, the maximum and minimum were identified.
- Lastly the pairs of frequency values that produced maxima were graphed in semi logarithmic scale.

The table that contains the group of obtained values of the chart is shown next (Table 1).

Table 1. Average of the goals obtained in 10 executions of 100 000 iterations, in 450 selected combinations (move, search), in a space of 500x300pixels and density of appearance of the pattern of 1%. Column S stands for *Search.*

S	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
10	78	72.5	68.7	66.2	54.1	50.5	53.3	47	45.9	25.4	21.7	20.5	18	18.7	18.2
20	76.8	94.1	100.1	99.8	97.3	99.4	95.8	96.1	88.4	46.6	44.9	40.5	35.4	33.5	30.9
30	62	85.5	98	102.4	102.3	108.2	110.2	105.1	111.8	60.2	58.8	54.6	51.6	48.6	46.3
40	51.9	78.1	89.4	99.4	103.8	106.5	107.6	108	109.4	69	71	65.7	60.5	59.3	55.4
50	40.8	67.2	80.4	91.6	95.3	105	105.1	105.3	110.2	75	73.3	72.4	70.8	68.2	63.4
60	36.9	63.2	78.5	87.7	94.1	96.9	100	107.3	104.5	81.9	86.8	75.6	71.6	72.4	66.3
70	30.9	57.8	69.6	79.2	88.6	92.5	97.6	104.5	104.9	87.1	86.2	84.3	81.2	78	74.6
80	28	50.6	67.2	80	84	90.4	95.7	98.7	99.6	94.2	89.6	87	84.9	77.5	76.9
90	25.7	48.6	62.4	72.6	80	87	92.7	95.3	96.6	96.2	92.2	92.67	85.5	83.7	83.7
100	40.64	88.45	105.45	108.91	109.64	105.45	107.73	104.09	102.55	95.36	95	92.27	92.36	88.18	87
110	39.18	83.64	108.64	108.27	110.91	106.36	107.55	101	102.82	102.27	97.36	92.73	92.45	90	86.82
120	40.27	88.55	107	107.82	111.45	108.91	109.45	104.18	101.73	99.73	98.55	98.91	95.18	96.18	84.18
130	38.27	88.18	109.91	105.55	112.91	111.36	107.82	106.18	103.45	105.91	102.09	97.64	97.73	95.82	92.45
140	38.36	78.55	104.27	111.27	113.36	114.18	108.27	108.82	103.82	105.45	102.09	98.91	99.36	95	94.55
150	37.36	75.27	98.27	110.91	112.45	112.73	108.73	107.27	108.45	105.27	103.36	97.64	102.55	95.91	97.82
160	33.22	82.22	95.56	108.22	109.11	111.67	110.78	110.78	109.00	103.67	101.78	101.20	103.30	100.50	100.10
170	36.00	71.44	97.56	111.22	112.00	113.33	107.44	108.44	112.44	107.89	102.78	105.00	102.40	101.40	99.80
180	30.56	81.00	101.89	109.00	107.22	115.11	114.11	105.44	109.00	107.89	103.67	103.90	98.60	101.20	100.00
190	31.00	71.56	91.44	109.00	111.67	115.00	110.33	108.11	109.89	106.56	108.67	105.60	105.20	101.90	96.10
200	31.33	69.89	96.33	109.56	110.78	111.00	115.33	112.22	108.78	108.67	105.44	107.30	96.90	104.10	99.80
210	26.11	74.33	92.56	107.67	108.89	110.11	111.00	108.89	108.00	112.22	108.78	109.90	104.60	104.30	104.00
220	31.67	62.56	99.78	102.33	107.78	109.56	114.89	112.56	111.11	108.78	108.56	109.10	106.30	104.20	102.70
230	26.11	70.89	91.56	105.11	111.00	115.78	112.11	112.56	107.56	110.56	110.33	108.70	108.00	104.10	103.20
240	27.78	64.67	89.00	110.67	109.67	115.33	108.67	109.56	114.56	109.22	110.67	109.10	107.30	105.10	103.00
250	20.00	60.22	87.11	108.78	114.33	112.56	111.89	112.56	113.67	111.22	111.90	109.40	108.90	105.20	99.90
260	27.1	70	86.9	108.4	110.6	112.7	111.8	111	108.8	109.3	111.8	109.3	108.2	109.3	105.9
270	29.6	67.7	91.6	99.7	111.7	109.7	111.9	115.6	111.1	110	111.7	109.8	111.2	109.2	108.9
280	27.9	59.5	98.2	101.6	111.8	111.3	110.4	110.8	109.8	110.1	110.7	110.8	113.4	103.7	105.3
290	24.7	53.6	87.7	105.6	99.2	113.3	111.1	112.4	113.8	112.8	111.5	109.1	109.9	110.1	110.9
300	27.9	69.2	92	103.1	107.2	112.4	110.3	113.6	113.1	115.5	110.7	110.2	110.2	109.2	107.8

In Table 1 two max values have been shadowed in the local search address (sniffing) for each advance value. Were selected two max values for each advance value. First max values were shadowed in dark grey, second in light grey. Max values except first and second column, maintain a large distance from the chart beginning. When advance increases they come closer each other more and more. Max values can be identified more easily in the surface graph 3.5 like two mountains in light grey.

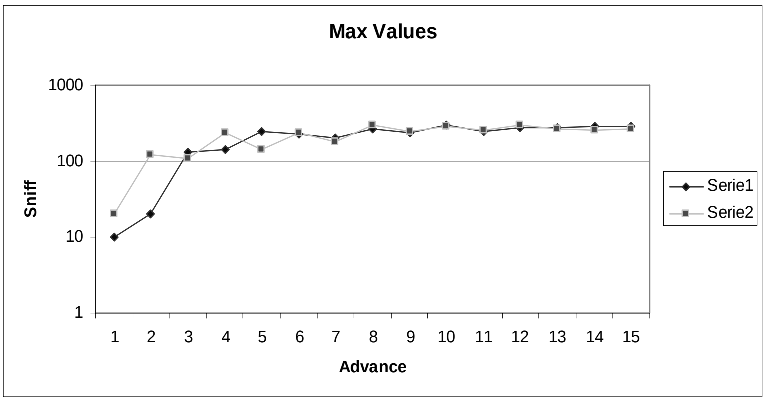


Fig. 3. Finding max values in spaces of 500x300 pixels with densities of pattern appearance to 1%.

In Fig. 4 the combinations that produced maxima have been graphed. We can observe a behavior that resemblance to an exponential one.

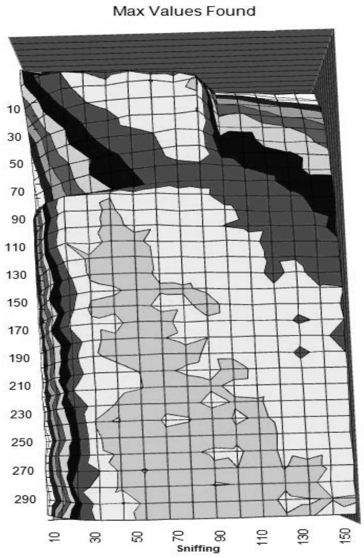


Fig. 4. Relationship of the T$_{move}$ and T$_{search}$ values that generated maxima in spaces of 500x300pixels with densities of pattern appearance to 1%.

In the following experiment, again it has been used a sample of 450 advance combinations and sniffing in a space of 600x400 pixels, with the purpose of assuring that the behavior observed in the previous experiment stays.

3.2 Finding Goals in a Space of 600x400 Pixels

- It was designed a pattern of 5x2pixels inserted in a 600x400pixels space with a density to 1%.
- In the previous space, it was executed an agent with a combination (moving and searching values) selected.
- The moving and searching values are in a range from 10 to 150 units for the advance and of 10 to 300 units for the sniff. The combinations were formed with 10 distance units, in the way (10,10), (10,20), (10,30), etc.
- For each combination of values (move, search) that form the sample, the goals were averaged found by an agent in 10 executions.
- Each execution had a duration of 100 000 iterations.
- Then a chart was built with the number average of goals found by the agent in each one of the combinations (move, search).
- In the previous chart, the maxima and minima are identified.
- Lastly the pairs of frequency values that produced maxima were graphed in semi logarithmic scale.

Table 2. Average of the goals obtained in 10 executions of 100 000 iterations, in 450 combinations (move, search), in a 600x400pixels space and density of pattern appearance to 1%. Column S stands for *Search.*

S	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
10	87.7	79.4	79.0	76.9	59.7	51.7	51.4	52.6	45.3	26.0	21.8	21.3	18.0	17.8	17.1
20	89.9	109.9	114.0	116.8	117.7	117.6	108.2	107.2	101.0	48.8	43.1	41.1	38.6	35.9	33.0
30	73.3	96.5	117.5	122.3	124.2	133.1	126.6	133.0	124.2	65.9	59.3	60.0	54.0	49.4	48.4
40	64.2	86.0	111.7	116.2	124.0	137.2	133.9	135.7	130.1	78.0	74.1	73.7	65.7	60.8	59.5
50	60.8	79.1	98.0	112.1	119.7	123.3	131.3	130.7	138.7	84.8	82.9	74.1	72.2	73.8	67.9
60	54.2	69.3	90.4	106.1	113.3	121.5	126.9	126.8	132.5	100.5	92.4	86.2	85.4	80.6	80.8
70	49.1	62.5	81.9	95.1	103.1	117.0	125.7	119.7	126.6	107.5	102.1	92.9	91.4	82.3	82.4
80	54.9	58.2	74.4	90.7	103.3	106.3	114.2	121.5	124.2	107.9	103.4	101.9	97.7	93.0	87.9
90	45.9	52.0	67.2	84.4	93.6	102.9	108.4	114.4	123.7	118.7	109.0	101.8	100.1	101.4	96.6
100	42.8	95.6	120.8	135	144.3	136.5	130.8	128.3	120.8	117.3	115.4	108.4	106.5	103	97.4
110	40.5	90.8	128.3	126.3	134.8	143.8	135.7	126.4	125.5	126.2	117.9	115.7	110.4	111.2	103.4
120	38.9	89.6	134.1	143	139.1	141.7	138.5	128.3	132.2	128.5	117.7	113.8	121.8	110.1	113.8
130	37.2	98.9	121	130.9	139.7	140.4	134.5	132.3	131.8	129	122.5	121.2	122.1	112.5	112.7
140	34.6	81.3	118.7	136.7	137.1	143.7	139.4	136.7	132.4	134	130	126.3	123.4	118	113.7
150	35.9	93.3	115.4	141.2	139.6	145.7	144.8	138.6	136.4	134.5	126.6	125.2	123.1	120.9	122.8
160	35.2	80.7	122.4	132.5	146.3	147.8	140.3	143.3	136.4	133.5	129.1	130.3	123.3	120.9	115.6
170	32.9	89.7	119.1	136.5	144	140.3	144.7	144	140	137.4	134.3	130.9	129.4	124	124.9
180	38.2	75.4	115.1	133.7	147.1	150.9	146.2	139.2	144.4	140.4	133.9	128.2	128.4	127.6	121.9
190	32.2	72.9	114.9	122.7	140.5	144.9	140.9	141.9	143.4	141.2	139.8	131.7	133.5	128.4	121
200	28.9	76.3	113.2	134.9	139.2	144.2	149.6	144.2	140.2	140.5	135.1	130.4	133.5	135.7	128.2
210	32.6	72.4	115.8	135.9	142.1	148.6	144.8	144	145.5	138.7	135.9	133	132.3	128	128
220	25.9	68.7	116.6	137.9	145.9	145.5	146.9	145.5	142.6	140.5	143.5	133.6	136.2	130.3	131.2

230	31.3	73	113.6	138.8	150.1	147.9	149.3	144.1	144.4	142.4	141.3	141.8	133.6	133	133.7
240	28.8	75.4	108.6	143.2	136	144.9	146.5	151.9	146	144.4	139.6	143.3	135.5	135	130.9
250	23	68	106.1	133.8	136.2	146.4	153.7	146.7	148.6	143.3	140.2	142.2	137.3	136.1	127.9
260	28.1	69.2	103.5	136.4	150.7	147.2	149.5	142.3	147.8	142.9	141.8	138.4	133.3	135.9	133.4
270	28.2	67.8	104.2	141.6	146.9	148.1	146.5	149.2	145.3	145.8	141.6	141.4	139.1	137.5	135
280	24.2	61.9	106.4	124.9	133.9	148	149.9	151.5	144.4	146.2	136.1	143.1	140.8	139.7	135
290	26.9	65.8	106.8	124.3	147.2	138	151.1	145.1	152.7	151	142.9	139.6	140.2	142	136.2
300	25.8	69.9	102.7	121.8	125.9	147.1	143.5	154.5	144.6	144.2	146.1	143.4	141.4	141.9	137.7

The same result was found in both experiments (3.1and 3.2). In Table 2 two max values have been shadowed in the local search address (sniffing) for each advance value. We selected two max values for each advance value. First max values were shadowed in dark grey, second in light grey. Max values except first and second column, maintain a large distance from the chart beginning. When advance increases they come closer each other more and more. Max values can be identified more easily in the surface Fig. 5 like two mountains in light color.

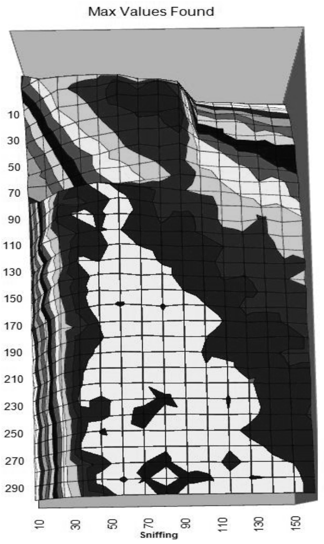


Fig. 5. Relationship of the T_{move} and T_{search} values that generated maxima in spaces of 600x400 pixels with densities of pattern appearance to 1%.

In Fig. 6 the combinations that produced maxima have been graphed. We can observe a behavior that resemblance to an exponential one.

As it is shown in Fig. 4 and Fig. 6, the moving and searching combinations that generate the maximum probability of goals, move from a similar way to an exponential assuring that the phenomenon is of a complex type.

In the following section, the effect of the space size when finding patterns is studied. For this, several spaces of different sizes were designed and in each one an agent was executed with a single moving and searching combination (20, 20), (30, 125), (50, 40) and (70, 30). The result is shown next.

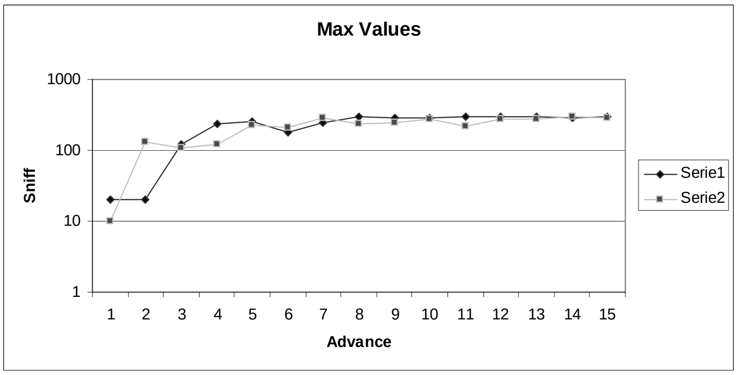


Fig. 6. Relationship of the T_{move} and T_{search} values that generated maxima, in spaces of 600x400pixels with densities of pattern appearance to 1%.

3.3 Maximum Goal Probability according to the Space Size

- Was designed a pattern of 5x2pixels inserted in spaces of different sizes with a density to 1%. The spaces were squared (same height and width).
- In the previous space an agent was executed with a combination of moving and searching values of (20, 20), (30, 125), (50, 40) and (70, 30). Those values were selected because they produced max values in previous tests. Those values appear like restrictions on resources used by the agent.
- For each space and combination 10 agents in an independent way were executed and the goals were averaged among all.
- Each execution had a duration of 100 000 iterations.
- A chart was built with averages of goals found by agent in each space.
- Knowing the total number of patterns inserted in the space, the percentage of goals/patterns was calculated with the purpose of knowing the average space that an agent sweeps with the selected combinations of advance and sniff.
- Lastly the obtained percentages were graphed.

The spaces were designed in several sizes, starting in 50 square pixels and stopping in 300 square pixels. Look that the larger space, the fewer goals found. Although the max goals number was in 50 and 75 pixels, these values were eliminated because it is easy to find goals in so little spaces. Then the value obtained like a maximum was 125pixels.

The results of the experiment are shown in table 3. The column X*X contains the size in square pixels of the space. Column x contains side size in pixels (local search window). The column #goals, contains the number of goals found by the agent. The column #patterns, contains the number of total patterns inserted in the space. Lastly, the column goals/patterns contain the percentage of the goals/patterns that was able to find the agent during the ten executions in each space.

Table 3. Average of the goals obtained in 10 executions of 100 000 iterations, with the moving and searching combination a) (20,20), b) (30, 125), c) (50, 40) and d) (70, 30) in different sizes spaces and density of pattern appearance to 1%.

a)

(20, 20)

X*X	X	#goals	#patterns	goals/ patterns
2500	50	4	4	100.00
5625	75	6.5	7	92.86
10000	100	11.5	15	76.67
15625	125	14.4	18	80.00
22500	150	10.5	15	70.00
28900	170	19.3	27	71.48
30625	175	26.3	38	69.21
32400	180	25.8	38	67.89
40000	200	18.4	32	57.50
62500	250	26.2	62	42.26
90000	300	27.9	83	33.61

b)

(30 , 125)

X*X	X	#goals	#patterns	goals/ patterns
2500	50	4	4	100.00
5625	75	6.6	7	94.29
10000	100	11	15	73.33
15625	125	14.1	18	78.33
22500	150	11.4	15	76.00
28900	170	19	27	70.37
30625	175	28.7	38	75.53
32400	180	28.4	38	74.74
40000	200	21	32	65.63
62500	250	33.5	62	54.03
90000	300	39.6	83	47.71

c)

(50, 40)

X*X	X	#goals	#patterns	goals/ patterns	
2500	50	2.85	4	71.25	
5625	75	6.2	7	88.57	
10000	100	11.4	15	76.00	
15625	125	13.7	18	76.11	
22500	150	10.8	15	72.00	
28900	170	17.4	27	64.44	
30625	175	24.7	38	65.00	
32400	180	24	38	63.16	
40000	200	18.4	32	57.50	
62500	250	26.9	62	43.39	%
90000	300	27.1	83	32.65	%

d)

(70, 30)

X*X	X	#goals	#patterns	goals/ patterns
2500	50	1.125	4	28.13
5625	75	4.57	7	65.29
10000	100	8.9	15	59.33
15625	125	11.2	18	62.22
22500	150	8.9	15	59.33
28900	170	15	27	55.56
30625	175	20.3	38	53.42
32400	180	18.4	38	48.42
40000	200	14.6	32	45.63
62500	250	19.2	62	30.97
90000	300	21.2	83	25.54

The combinations (20,20) and (30, 125) were chosen being maximums in the tests studied in the previous section. The values (50,40) and (70, 30) too were chosen being maximums but in tests that were not considered for this article. These tests were similar to the previous ones but with a small number of combinations (135 instead of the 450 combinations of test 3.1 and 3.2).

According to the previous result, an agent with capacities similar to the exposed ones in these tests can cover a space of 125 square pixels with the maximum goal probability.

4 Conclusions and Future Works

In this work, we tried to simulate a metaphor of jumping behavior like a search algorithm that an agent uses to find 2-D patterns. Every cycle, the agent is alternating periods of moving and searching (local search). Moving and searching times supposed as restrictions in the resources of a search agent. Therefore, the best results are those that combined maximum number of found patterns (goals) and a minimum of resources.

A large advance frequency makes that the agent goes far away distances before making a search. Big sniffing frequencies (local search), makes the agent execute local searches in large areas. Small advance frequencies will make the agent to take more time in exploring large search spaces. And finally, small sniffing frequencies will not detect the pattern, if this one is bigger than the window (that the agent explores during the sniffing time).

The results showed that the max probability of goals is when the advance frequency is similar or smaller than the sniffing frequency (when the agent has more time for local search than exploring the space). The experiments showed that exist two sniffing values for each moving frequency that generate a maximum (of goals). This agrees according to [2].

The relationship among frequencies (movement and sniff) that generates greater probabilities of finding patterns showed an exponential form, which assures that the nature of the exposed phenomenon is of a complex type.

Actually we use this metaphor to simulate the behavior of several agents looking for patterns in a selected space with and without communication. Results will be shown in next work.

References

1. Kramer, D.L., Mclaughlin. R.L: The Behavioral Ecology of Intermittent Locomotion. AMER. ZOOL., 41, pp. 137-153 (2001)
2. Benichou, O., Coppey, M., Moreau, M., Suet, P-H., Voituriez, R.: Optimal search strategies for hidden targets. Physical Review Letters 94, 198101, pp. 1-4 (2005)
3. Grver, L.: A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th ACM Symposium on the Theory of Computing, pp. 212-219 (1996)
4. Boyer, M., Brassard, G., Hoyer, P., Tapp, A.: Thight bounds on quantum searching. In: Workshop of Phys. Comp. 96, pp. 36-43 (1996)
5. Lorenz, C., Mottl, P., Wattenhofer, R.: Ant Algorithm. Distributed Computing Group, Thesis, Swiss Federal Institute of Technology Zurich.

A Branch and Bound Algorithm for Ternary Covering Arrays Construction using Trinomial Coefficients

Jorge Martinez-Pena and Jose Torres-Jimenez

CINVESTAV-Tamaulipas, Information Technology Laboratory, Km. 5.5 Carretera
Cd. Victoria-Soto la Marina, 87130, Cd. Victoria Tamps., Mexico
{jmartinez,jtj}@tamps.cinvestav.mx

Abstract. A ternary covering array (CA$(N;t,k,3)$) is an $N \times k$ array with entries from the set $\{0,1,2\}$ where every $N \times t$ subarray contains each of the 3^t combinations at least once. Here, t is called the strength, k the degree or number of factors, and the alphabet is set to 3. Covering arrays (CAs) are combinatorial designs that have applications in experimental designs and they have been proved to be useful in software and hardware testing. This paper presents a Branch & Bound algorithm for constructing small instances of ternary CAs using trinomial coefficients. As a result, we derived direct constructions for some ternary CAs.

Keywords: Design of Experiments, Ternary Covering Arrays, Branch & Bound, Trinomial Coefficients.

1 Introduction

Combinatorial designs have a wide use in the design of experiments. An orthogonal array (OA) is an example of a combinatorial object that has been applied successfully in this area [1]. An OA, denoted by OA(t,k,v) is an $N \times k$ array with entries from a set of v symbols such that each $N \times t$ subarray contains exactly once the v^t different combinations. The parameter t is called the strength, k is the number of factors, and v is called the alphabet. However, the orthogonal arrays are very restricted objects because they do not exist for many configurations of the parameters t, k and v. Suppose that an experiment requires an interaction of size 6 among 8 factors, with 3 symbols per factor, it is not possible to construct an OA$(6,8,3)$. Without the required OA, an exhaustive experiment should be performed and would require 6561 tests.

A less restricted combinatorial object is a covering array (CA). A CA exists for any possible combination of parameter values t, k and v. A CA [2] is defined as follows:

Definition 1. *Let t be the strength, k the degree, and v the order (or alphabet), with $2 \leq t \leq k$ and $v \geq 2$. A covering array, denoted by CA$(N;t,k,v)$, is an $N \times k$ array that has entries from the set $X = \{0,..,v-1\}$ such that for every $N \times t$ subarray each of the v^t combinations are contained at least once.*

In this way, a ternary CA is a CA with v set to three. For instance, a CA$(N; 6, 8, 3)$ can be constructed with 1177 experiments only (result obtained from our algorithm and taken from Table 1) which means savings around 82% off in experimental tests. The same problem arises in software and hardware testing. Recent experimental results have shown that CAs are an efficient solution for generating tests suites for interaction testing problems [3]. Due to the importance of constructing CAs with the minimum number of tests we begin to describe the CA construction (CAC) problem.

The covering array number (CAN(t, k, v)) represents the minimum number of rows for which a CA$(N; t, k, v)$ exists. A CA$(N; t, k, v)$ whose parameter $N =$ CAN(t, k, v) is called optimal [4].

The CAC problem consists in obtaining optimal covering arrays. Despite the existence of some special cases where the solution is obtained by polynomial order algorithms, the general case of determining the CAN is a hard combinatorial optimization problem [5]. Therefore, it is desirable to obtain covering arrays with minimum value of N, even when the CAN is not reached. Some of the best known solutions are presented in [6]. For viewing explicit covering arrays, there is a covering array repository [7] that maintains some optimal and near optimal CAs.

Branch and bound (B&B) algorithms allows exploring the whole search space of the problems using an intelligent approach to avoid looking for solutions that do not improve the current best solution [8]. B&B techniques [9] have been applied to the problem of constructing binary CAs. This paper presents a B&B algorithm for the construction of ternary CAs that uses the trinomial coefficients to represent the search space.

This paper continues as follows. Section 2 presents the relevant related work in the construction of ternary CAs. Section 3 describes the trinomial coefficients [10] and the way we used them to construct ternary CAs. Section 4 describes our B&B algorithm that uses trinomial coefficients to represent the search space for constructing ternary CAs. Section 5 exposes the performed experimentation, a comparison of the results obtained with our B&B algorithm to the results obtained with a state-of-the-art greedy method and a set of direct constructions derived from the analysis of trinomial coefficients usage in the formation of the ternary CAs produced by the B&B algorthm. Finally, Section 6 gives important conclusions of this research work.

2 Relevant Related Work

The CAC problem consists in obtaining CAs with minimum value N. Most configurations of parameters t, k and v make the CAC problem a hard combinatorial optimization problem. Due to the importance of obtaining optimal or near optimal CAs, several strategies have been applied to the CAC problem. In this section we give a brief state-of-the-art in the construction of ternary covering arrays including direct, greedy, metaheuristic, recursive, algebraic and exact methods.

Direct methods construct CAs in polynomial time. In 1952, Bush [11] reported a direct method for constructing optimal CAs that uses Galois finite fields obtaining all $\mathrm{CA}(v^2; 2, v+1, v)$ where v is a prime or a prime power. Then, he generalizes the algorithm for any strength $t \geq 2$. Another direct method that can construct some optimal CAs is named zero-sum [12]. Zero-sum leads to $\mathrm{CA}(v^t; t, t+1, v)$ for any $t > 2$. Note that the value of degree is in function of the value of strength.

Different greedy methods have been designed for constructing CAs. For instance, IPOG-F [13] has been applied for small values of v and t and as large as possible value of k. This algorithm has been able to find some of the best known values of CAN for large k. However, it did not obtain the optimal CAs reported by the previous direct methods. The IPO family are some of the first computational strategies used for constructing CAs and they have been updated since the first version of the IPO algorithm. Although the greedy approach has the guarantee of constructing a CA, values of N are generally higher in comparison to some non-deterministic search algorithms.

Several versions of metaheuristic algorithms have been implemented for the construction of CAs including works using Simulated Annealing (SA) [14, 15], Tabu Search [16] and those in the cathegory of artificial life algorithms such as Genetic Algorithms and Ant Colony Optimization [17]. SA is the best metaheuristic reported for constructing CAs from small to medium large values of k. For larger values of k, metaheuristics require exponential time for attempting to obtain optimal or near optimal results.

Other strategies for obtaining ternary CAs include recursive methods [18, 19]. These algorithms makes use of small CAs called ingredients in order to construct CAs with larger k. Most of best known values of CAN for large k values are constructed using recursive methods.

There are algebraic strategies that have been applied to the CAC problem. The *group construction* approach and *cyclotomic vectors* have in common the use of a particular vector which is submited to a set of operations, usually rotations and permutations, to generate all rows necessary for CA [20, 21]. However, algebraic methods have the premise that we have the requiered vector for that specific construction. In this sense, a previous search for the correct vector is required.

Finally, exact algorithms have not been exploited as other strategies due to the capacity of generating only small instances of CAs in reasonable time. Up to now, we could find only two papers related to exact algorithms, both of them using a B&B approach [22, 9]. These works only construct binary CAs and are based on symmetric breaks and isomorphism avoidance.

In this paper we use the results reported by IPOG-F as a benchmark.

3 Proposed Representation

This section describes the trinomial coefficients and how we use them to represent the search space in the construction of a ternary CA.

3.1 Trinomial Coefficients

Given a vector of k positions and three available values per position x, y and z, we can see a trinomial coefficient as the number of combinations with repetition obtained by using the k positions vector with a times the value x, b times the value y and c times the value z ($x^a y^b z^c$), where the sum of a, b and c equals k. A formal definition of trinomial coefficients [10] is:

Definition 2. *Let $0 \leq a, b, c \leq k$ and $k = a + b + c$. A trinomial coefficient, denoted by $\binom{k}{a,b,c}$, is the coefficient obtained by the next equation*

$$\binom{k}{a, b, c} = \frac{(a + b + c)!}{a!b!c!} \tag{1}$$

The trinomial coefficients arise in the expansion of a trinomial $(x+y+z)^k$. The solution of a trinomial expansion is the trinomial theorem [23] and is determined by the following identity.

$$(x + y + z)^k = \sum_{\substack{0 \leq a,b,c \leq k \\ a+b+c=k}} \binom{k}{a, b, c} x^a y^b z^c \tag{2}$$

The equation in 2 is very important. For any k, it describes all the possible trinomial coefficients. Moreover, the number of trinomial coefficients for a given k is $\binom{k+2}{2}$ and the sum of those trinomial coefficients is 3^k.

3.2 Representation

We use the trinomial coefficients for the representation of the search space in the construction of ternary CAs. It is clear that any CA is formed by a row set. In this sense, a trinomial coefficient represents a particular subset of rows which may belong to a ternary CA. First, we give some preliminar statements that lead to our proposed representation. Then, we describe our proposed representation using a clear example.

Definition 3. *Let $0 \leq a, b, c \leq k$, $k = a+b+c$ and $k \geq 2$, where k is the ternary CA degree. A candidate row subset $\mathscr{R}^k_{a,b,c}$ is a collection of rows obtained by the trinomial coefficient $\binom{k}{a,b,c}$ and its cardinality is equal to that coefficient. The candidate row subset is generated by evaluating all combinations using $0^a 1^b 2^c$ symbols, i.e. symbol 0 is used a times, symbol 1 is used b times and symbol 2 is used c times over a k-column row.*

The previous definition leads to the next theorem.

Theorem 1. *Let $\mathscr{A}$ be a set of k-th degree trinomial coefficients. For any strength $2 \leq t \leq k$, a vertical concatenation of the row subsets generated by each trinomial coefficient in $\mathscr{A}$ may construct any ternary CA.*

Proof. Let the strength and the degree of a required CA equal q. Let $\mathscr{C}$ be a $3^q \times q$ array. Adjoin the $\binom{q+2}{2}$ trinomial coefficients of q-th degree in the set $\mathscr{A}$. For each element in $\mathscr{A}$ generate its candidate row subset and append it vertically to $\mathscr{C}$. Then $\mathscr{C}$ is an optimal CA$(3^q; q, q, 3)$.

We verify the result is a ternary CA by looking into the definitions of the trinomial theorem and of the candidate row subsets. The trinomial theorem generates all possible trinomial coefficients of q-th degree. Remark that the sum of all q-th degree trinomial coefficients is 3^q. Hence if we transform each trinomial coefficient of q-th degree into candidate row subsets we will be producing 3^q different rows. These rows represent all the possible combinations that any $N \times q$ subarray must contain. By definition, any CA(q, q, v) has only one subarray of size q. Therefore, we have constructed an optimal ternary CA of strength q and degree q.

Now, consider the constructed CA$(3^q; q, q, 3)$. The strength value in a CA is upper bounded by degree value. For any $2 \leq s < q$ we automatically derive a CA$(3^q; s, q, 3)$. Then we can construct a ternary CA of any strength and any degree. Thus the theorem is correct.

Figure 1 summarizes our proposed representation by giving an explicit example of construction of a ternary CA. In this example, we construct a CA$(9; 2, 3, 3)$ by using a set $\mathscr{A}$ with 4 trinomial coefficients. In the middle, we can observe a table describing each trinomial coefficient in $\mathscr{A}$ and their corresponding candidate row subsets. The table to the right displays the array $\mathscr{C}$ (the ternary CA), which is composed of every row generated by $\mathscr{A}$.

$$\mathscr{A} = \{\tbinom{3}{3,0,0}, \tbinom{3}{0,3,0}, \tbinom{3}{0,0,3}, \tbinom{3}{1,1,1}\}$$

$$\mathscr{C} = (\mathscr{R}^3_{3,0,0} + \mathscr{R}^3_{0,3,0} + \mathscr{R}^3_{0,0,3} + \mathscr{R}^3_{1,1,1})$$

$$N = \sum \mathscr{A} = 9$$

a)

$\mathscr{A}$	$\mathscr{R}^k_{a,b,c}$
$\binom{3}{3,0,0}$	0 0 0
$\binom{3}{0,3,0}$	1 1 1
$\binom{3}{0,0,3}$	2 2 2
$\binom{3}{1,1,1}$	0 1 2
	1 2 0
	2 0 1
	0 2 1
	1 0 2
	2 1 0

b)

$\mathscr{C}$
0 0 0
1 1 1
2 2 2
0 1 2
1 2 0
2 0 1
0 2 1
1 0 2
2 1 0

c)

Fig. 1. Representation of a CA$(9; 2, 3, 3)$ in trinomial coefficients. a) The mathematical formulation. b) The candidate row subsets from trinomial coefficients. c) The produced ternary CA.

4 Branch and Bound Algorithm

We designed an exhaustive search algorithm that finds a ternary CA in the trinomial coefficient domain. The number of rows obtained is the minimum rows for which a ternary CA exists using our trinomial coefficients representation. The algorithm takes as input the strength t and the degree k for the construction of the ternary CA and it is structured in three steps.

In the first step we generate the set of all trinomial coefficients of k-th order and their representing row subsets in $\mathscr{T}$. The second step is a backtracking algorithm that implements a B&B technique for performance improvement. The backtracking step incrementally constructs a ternary CA by adding *feasible* elements from $\mathscr{T}$ into $\mathscr{A}$. We call *feasible* to a trinomial coefficient that is able to construct the ternary CA without exceeding the number of rows required by the current best ternary CA. Each time a solution improves the current best solution it is stored in $\mathscr{S}$ and it is used as an upper bound for the backtracking. Finally, the third step transforms $\mathscr{S}$ into the ternary CA and store it into an output file.

Figure 2 is a flow diagram that describes our B&B algorithm. The first step is only the first state in the diagram, while the third step is the last state where we output the ternary CA. As observed, the second step of the algorithm is the main part of this flow diagram. Note that each time a better solution is found, we update the upper bound and the current best solution. This upper bound determines which is a next feasible trinomial coefficient based on the sum of that coefficient and the current number of rows in $\mathscr{A}$. If this sum is longer than the current best upper bound, the trinomial coefficient is ignored and the algorithm goes to the next trinomial coefficient. Each time the algorithm joins the next feasible trinomial coefficient to $\mathscr{A}$, a partial test CA is performed.

A partial test CA requires a $\binom{k}{t} \times 3^t$ matrix. For each combination of all subarrays, this matrix allows to decide how many combinations are missing to complete the ternary CA. We perform the partial test CA in $\binom{k}{a,b,c}\binom{k}{t}$. In this sense, each row of a candidate row subset is verified by this partial test.

5 Experimentation and Results

We implemented the B&B algorithm described in section 4. The algorithm was coded in C and compiled with *gcc*. This section presents the comparison between the results from IPOG-F and the results obtained with our B&B algorithm that uses trinomial coefficients. As a result of the experimentation, some direct constructions for ternary CAs are described using specific trinomial coefficients.

5.1 Results from B&B Algorithm

We run the experimentation for different values of strength and degree. Strength ranged in $2 \leq t \leq 6$ and degree ranged in $t + 1 \leq k \leq 15$. Results for strength 5

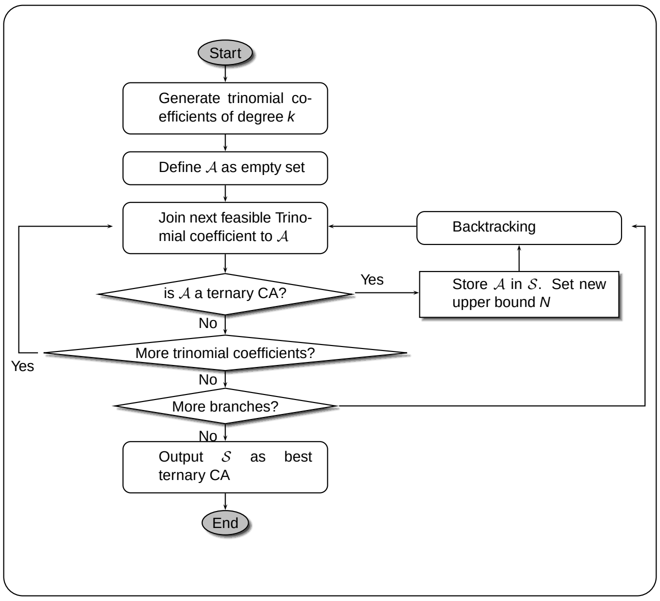

Start
Generate trinomial coefficients of degree k
Define $\mathcal{A}$ as empty set
Join next feasible Trinomial coefficient to $\mathcal{A}$
Backtracking
is $\mathcal{A}$ a ternary CA?
Yes
Store $\mathcal{A}$ in $\mathcal{S}$. Set new upper bound N
No
More trinomial coefficients?
Yes
No
More branches?
No
Output $\mathcal{S}$ as best ternary CA
End

Fig. 2. Flow diagram of our B&B algorithm.

and strength 6 are not complete due to the long time the algorithm requires to finish that experimentation.

Figure 3 displays the behaviour of results obtained with our *B&B* algorithm. We can observe that the growth of N using strength 2 is lineal in k. For strengths 3 and 4, the growth of N is quadratic in k. Finally, the growth of N for strength 5 and 6 is cubic in k.

5.2 Comparison Between IPOG-F and our B&B Algorithm

We compared the results obtained from the greedy algorithm IPOG-F [13] to the results from our B&B algorithm. The comparison is done using the vales of N from the constructed CAs.

Table 1 concentrates the results using strength values in range $2 \leq t \leq 6$ and degree values in range $t+1 \leq k \leq 15$. We can observe that our B&B algorithm outputs better results for any $k \leq t+2$. However, the rest of cases IPOG-F offers better results than those obtained with our B&B. Notice that any result obtained with our *B&B* algorithm when $k = t+1$ equals the CANs. Results for strength 5 and strength 6 are not complete due to the long time the algorithm requires to finish that experimentation.

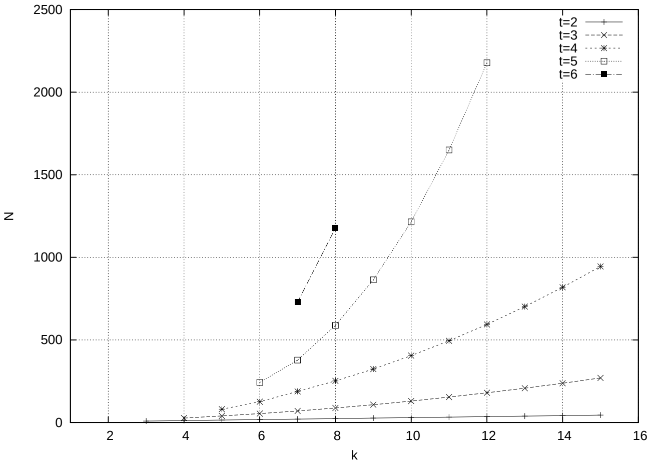


Fig. 3. Growth behaviour of N using our B&B algorithm. Abscissa values are the degree values for $2 \leq k \leq 15$ while ordinate values are the number of rows for the produced ternary CAs.

Table 1. Solution comparison between the IPOG-F and our B&B algorithm. Values of IPOG-F are in columns marked γ while the results obtained with our B&B algorithm are marked β. Bold values mean improved results.

k / t	2		3		4		5		6	
	γ	β	γ	β	γ	β	γ	β	γ	β
3	10	**9**								
4	12	**12**	34	**27**						
5	13	15	42	**40**	98	**81**				
6	15	18	49	54	140	**126**	318	**243**		
7	15	21	52	70	164	189	467	**378**	990	**729**
8	15	24	56	88	188	252	557	588	1490	**1177**
9	17	27	62	108	211	324	652	864		
10	19	30	66	130	228	405	738	1215		
11	19	33	68	154	248	495	815	1650		
12	20	36	71	180	262	594	885	2178		
13	20	39	76	208	277	702				
16	20	42	77	238	288	819				
15	20	45	80	270	302	945				

5.3 New Ternary Covering Array Direct Constructions

We derived some direct constructions by analyzing the trinomial coefficients formation for the ternary CAs found by our B&B algorithm.

These direct constructions match the results obtained in Table 1 and are able to construct more ternary CAs than those reported in our experimentation. In this sense, these constructions leads to ternary CAs that have the minimum possible value of N in our trinomial coefficient representation. Note that this value of N is not the CAN.

The following 4 theorems can construct ternary CAs by using a polynomial direct method. Each theorem focus the construction based in the strength of the generated ternary CAs.

Theorem 2. *$CA(3k;2,k,3)$ is directly derived using trinomial coefficients for any $k \geq 3$.*

Proof. Let $\mathscr{A} = \{\binom{k}{k-1,0,1}, \binom{k}{1,k-1,0}, \binom{k}{0,1,k-1}\}$ and any $k \geq 3$. Construct an array of size $3k \times k$ called $\mathscr{C}$. Concatenate vertically into $\mathscr{C}$ each of the candidate row subsets produced by $\mathscr{A}$. The result is a strength two ternary CA of degree k with optimal N in our trinomial coefficients representation.

Theorem 3. *$CA(k^2+3k;3,k,3)$ is directly derived using trinomial coefficients for any $k \geq 5$.*

Proof. Let $\mathscr{A} = \{\binom{k}{k-1,0,1}, \binom{k}{k-1,1,0}, \binom{k}{1,0,k-1}, \binom{k}{0,1,k-1}, \binom{k}{1,k-2,1}\}$ and any $k \geq 5$. Construct an array of size $(k^2+3k) \times k$ called $\mathscr{C}$. Concatenate vertically into $\mathscr{C}$ each of the candidate row subsets produced by $\mathscr{A}$. The result is a strength three ternary CA of degree k with optimal N in our trinomial coefficients representation.

Theorem 4. *$CA(4.5k^2-4.5k;4,k,3)$ is directly derived using trinomial coefficients for any $k \geq 7$.*

Proof. Let $\mathscr{A} = \{\binom{k}{k-2,0,2}, \binom{k}{0,2,k-2}, \binom{k}{k-2,2,0}, \binom{k}{k-2,1,1}, \binom{k}{1,1,k-2}, \binom{k}{1,k-2,1}\}$ and any $k \geq 7$. Construct an array of size $(4.5k^2-4.5k) \times k$ called $\mathscr{C}$. Concatenate vertically into $\mathscr{C}$ each of the candidate row subsets produced by $\mathscr{A}$. The result is a strength four ternary CA of degree k with optimal N in our trinomial coefficients representation.

Theorem 5. *$CA(\frac{4.5(k-1)^3-4.5(k-1)^2}{3};5,k,3)$ is directly derived using trinomial coefficients for any $k \geq 7$.*

Proof. Let $\mathscr{A} = \{\binom{k}{k-2,0,2}, \binom{k}{0,2,k-2}, \binom{k}{2,k-2,0}, \binom{k}{k-3,2,1}, \binom{k}{2,1,k-3}, \binom{k}{1,k-3,2}\}$ and any $k \geq 7$. Construct an array of size $(\frac{4.5(k-1)^3-4.5(k-1)^2}{3}) \times k$ called $\mathscr{C}$. Concatenate vertically into $\mathscr{C}$ each of the candidate row subsets produced by $\mathscr{A}$. The result is a strength five ternary CA of degree k with optimal N in our trinomial coefficients representation.

6 Conclusions

In this paper we presented a backtracking algorithm for the ternary CAC problem that makes use of trinomial coefficients as a representation of the search space. A B&B technique and partial testing CA were implemented for improving the performance in our algorithm.

Although the proposed algorithm in this research work takes much time using longer values of k and t, it proposes a new approach for modeling the search space. The proposed representation of search space based on trinomial coefficients was proved to be complete for constructing any ternary CA.

The results of our B&B algorithm were compared with reported results of the IPOG-F algorithm. Although most of the cases IPOG-F offered better N values, we can obtain better ternary CAs for any $2 \leq t \leq 6$ and any $k \leq t + 2$.

According to our experimentation, we conclude that the use of our representation of trinomial coefficients can lead to good ternary CAs for any strength t and any $k \leq t + 2$.

The analysis over the trinomial coefficient usage for the performed constructions using our B&B algorithm derived new direct constructions that uses specific trinomial coefficients for strengths 2 to 5. These direct constructions requires polynomial time and space thus there is no need to search those ternary CAs with our B&B algorithm.

We believe that our representation of trinomial coefficients for the construction of ternary CAs can lead to more general direct constructions than those reported in this paper. It is necessary to do more experimentation for higher strengths and a deeper analysis in the structure of the trinomial coefficients requiered for the generated ternary CAs.

Acknowledgements. This research work was partially funded by the following projects: CONACyT 58554-Cálculo de Covering Arrays and 51623-Fondo Mixto CONACyT y Gobierno del Estado de Tamaulipas.

References

1. Hedayat, A., Sloane, N.J.A., Stufken, J.: Orthogonal Arrays, Theory and Applications. Springer-Verlag, New York, US (1999)
2. Sloane, N.: Covering arrays and intersecting codes. Journal of Combinatorial Designs **1** (1993) 51–63
3. Kuhn, R., Lei, Y., Kacker, R.: Practical combinatorial testing: Beyond pairwise. IT Professional **10** (2008) 19–23
4. Ji, L., Yin, J.: Constructions of new orthogonal arrays and covering arrays of strength three. Journal of Combinatorial Theory, Series A **117** (2010) 236–247
5. Colbourn, C.: Combinatorial aspects of covering arrays. Le Matematiche (Catania) **58** (2004) 121–167
6. Colbourn, C.: Covering array tables. `http://www.public.asu.edu/~ccolbou/src/tabby/catable.html` (accessed Jun 1, 2010)

7. Torres-Jimenez, J.: Covering arrays repository. `http://www.tamps.cinvestav.mx/\~jtj/CA.php` (accessed Jun 1, 2010)
8. Kreher, D.L., Stinson, D.R.: Combinatorial algorithms: generation, enumeration, and search. SIGACT News **30** (1999) 33–35
9. Bracho-Rios, J., Torres-Jimenez, J., Rodriguez-Tello, E.: A new backtracking algorithm for constructing binary covering arrays of variable strength. In: MICAI. (2009) 397–407
10. Keeney, R.: On the trinomial coefficients. Mathematics Magazine **42** (1969) 210–212
11. Bush, K.: Orthogonal arrays of index unity. Annals of Mathematical Statistics **23(3)** (1952) 426–434
12. Colbourn, C., Dinitz, J.: The CRC Handbook of Combinatorial Designs. CRC press, Boca Raton (1996)
13. Forbes, M., Lawrence, J., Lei, Y., Kacker, R.N., Kuhn, D.R.: Refining the in-parameter-order strategy for constructing covering arrays. Journal of Research of the National Institute of Standards and Technology **213** (2008) 287–297
14. Cohen, M.B., Colbourn, C.J., Ling, A.C.H.: Augmenting simulated annealing to build interaction test suites. In: ISSRE. (2003) 394–405
15. Lopez-Escogido, D., Torres-Jimenez, J., Rodriguez-Tello, E., Rangel-Valdez, N.: Strength two covering arrays construction using a sat representation. In: MICAI. (2008) 44–53
16. Nurmela, K.J.: Upper bounds for covering arrays by tabu search. Discrete Appl. Math. **138** (2004) 143–152
17. Shiba, T., Tsuchiya, T., Kikuno, T.: Using artificial life techniques to generate test cases for combinatorial testing. In: COMPSAC '04: Proceedings of the 28th Annual International Computer Software and Applications Conference, Washington, DC, USA, IEEE Computer Society (2004) 72–77
18. Martirosyan, S., van Trung, T.: On *t*-covering arrays. Des. Codes Cryptography **32** (2004) 323–339
19. Hartman, A., Raskin, L.: Problems and algorithms for covering arrays. Discrete Mathematics **284** (2004) 149–156
20. Meagher, K., Stevens, B.: Group construction of covering arrays. Journal of Combinatorial Designs **13** (2005) 70–77
21. Colbourn, C.J.: Covering arrays from cyclotomy. Designs, Codes and Cryptography **55** (2010) 201–219
22. Yan, J., Zhang, J.: A backtracking search tool for constructing combinatorial test suites. J. Syst. Softw. **81** (2008) 1681–1693
23. Graham, R.L., Knuth, D.E., Patashnik, O.: Concrete Mathematics: A Foundation for Computer Science. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1994)

Iris Segmentation using Geometrical Methods

Gustavo L. Vidal-González[1], Alejandro A. Ramírez-Acosta[2],
Mireya S. García-Vázquez[1], and Ciro A. Martínez García-Moreno[1]

[1] Centro de Investigación y Desarollo de Tecnología Digital - CITEDI,
Av. del Parque No. 1310, Mesa de Otay,
Tijuana, Baja California, 22510,
Mexico
[2] MIRAL R&D, Imperial Beach,
USA
{vidal, mgarcia, ciro}@citedi.mx, ramacos10@hotmail.com

Abstract. In the field of biometric identification, iris recognition systems employ techniques to isolate the iris region to speed up the following stage of identification. Many existing iris segmentation techniques rely heavily on image operations to obtain high levels of accuracy but introduce unneeded computational overhead. Others offset the wide use of image operations by analyzing the image prior to attempting segmentation while maintaining a good deal of accuracy. Fewer still, offset the use of image operations even further by adopting fast deterministic approaches to the iris segmentation problem based on prior knowledge of the geometry of the iris, but achieve only modest accuracy. We propose a method for iris segmentation that focuses on minimum computational overhead while maintaining a good deal of accuracy which utilizes: pre-segmentation techniques based on image histogram analysis, canny edge detection image operations and finally a geometrical analysis to segment the iris region.

Keywords: Iris Localization, Circle Detection, Canny Edge Detection, Histogram Analysis, Iris Biometrics.

1 Introduction

The use of biometric identification systems has extended rapidly, due to the need of security systems that allow entry to access points based on individual verification. It is well known that biometric identifiers are of great use for individual human recognition. Thus, individual features such as fingerprints, iris textures, voice and etcetera are used as some of the more popular biometric identifiers for these security verification systems.

These related technological advances have been of great help for biometric identification because security systems employ a complex infrastructure that integrates a great deal of hardware as well as software to accomplish the task of identification. The present work utilizes identification through iris biometrics; it is worth mentioning that our area of interest is in the software of biometric identification. It is based on two related previous works that have applied quick

© G. Sidorov et al. (Eds.)
Advances in Soft Computing Algorithms
Research in Computing Science 49, 2010, pp. 73-84

deterministic approaches [1] (Liu *et al.*) and more accurate thorough approaches [2] (Sun *et al.*) to iris segmentation. This article presents a method of improvement in the detection of the iris by leveraging part of the accuracy of Sun *et al.* [2] using pre-segmentation techniques through histogram analysis and then proceeding with quick iris segmentation methods proposed by Liu *et al* [1]. This allows us to work with a standard set of grayscale iris images like the “NIR still eye image” set from the MBGC (Multi-Biometric Grand Challenge) [3].

The method defined in this article uses a pupil pre-segmentation technique by means of a threshold obtained from a histogram analysis of the original image [2]; Canny Edge Detection [4], to obtain pupil/iris contours; Traditional geometric methods to obtain the pupil center and pupil/iris radius [1]; and finally an outward scan starting from the pupil/iris radius to find the iris/sclera boundary [1].

This paper is organized as follows, in section 2 we discuss pre-segmentation through histogram analysis, in section 3 we show the result of obtaining contours using Canny edge detection, in section 4 we locate the pupil center as well as the pupil/iris boundary using geometrical methods, the modeling of a parametric circular motion equation is explained for the purpose of locating the iris/sclera boundary in section 5, in section 6 we discuss the results of the iris segmentation and finally we bring to light some conclusions of using this method in section 7.

2 Pre-segmentation Based on Histogram Analysis

Geometrical methods rely on obtaining genuine circle contours along the pupil/iris boundary after applying Canny edge detection. However, when the eye is not delimited in the image, more contours may be detected, be it from eye lashes, eyelids, eye brows, wrinkles, scars and etcetera. Therefore steps should be taken to facilitate the localization of the pupil/iris boundary. For example: an image acquisition system where the user’s eye closes-in into the camera with appropriate lighting [1], thus producing an image that is easy to work with; the method explained in this article assumes a subpar image with eyelashes, eyelids and eyebrows included in the image.

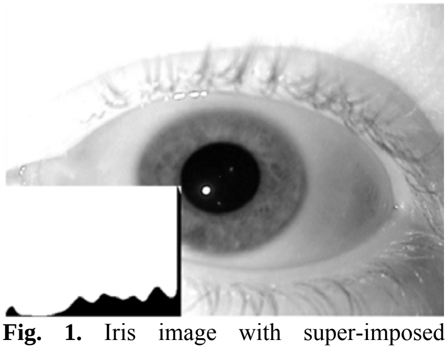


Fig. 1. Iris image with super-imposed Intensity histogram (*H*).

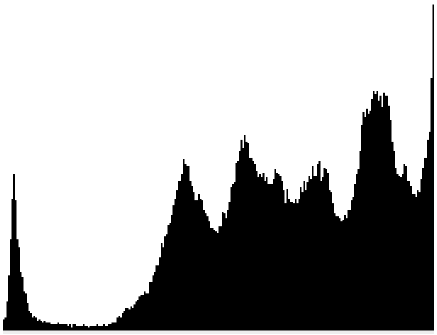


Fig. 2. Close up of Intensity histogram (*H*).

Another technique for delimiting the pupil region is applying a dark intensity threshold based on a histogram analysis. Being that the darkest grayscale intensities are mostly located in the region of the pupil, the threshold provides coarse pupil segmentation therefore eliminating many unwanted objects in that image [2].

Given that the threshold value should be carefully considered, we propose a method consisting of low pass filtering and derivative like operations on the histogram to faithfully locate a good threshold value from the histogram.

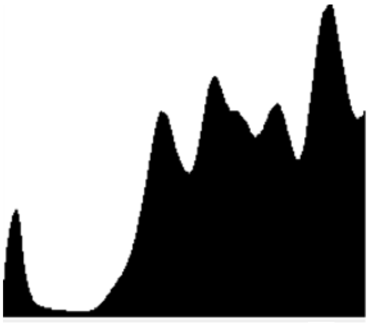


Fig. 3. Average filtered histogram ($\overline{H}$).

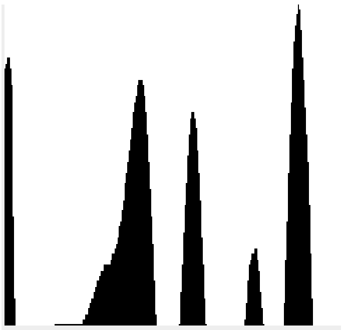


Fig. 4. Pseudo-derived histogram ($\dot{H}$)

The detailed method for obtaining the appropriate threshold value is described as follows:

1) Pixels of image are read in standard 8 bit grayscale format. The histogram (*H*) is calculated by surveying all the grayscale pixel intensities in the iris image and storing the occurrence of the intensities in 256 (2^8) possible bins. If quantization of intensities by means of bin reduction (i.e. 128, 64, 32 bins) would be used it would hinder the ability of the process to pinpoint a needed single grayscale intensity/bin. When the image histogram is plotted the darkest intensities are the left most values in the histogram corresponding to the near-zero most values and the brightest intensities are the right most values in the histogram (as seen in figure 1) (only positive values have been plotted).

2) Upon closer inspection of the raw histogram (H) (see figure 2), trends in the form of valleys and peaks could be seen with each one having their own frequency. However, if a valley or peak detection routine was to be applied to the raw histogram, an incorrect point maybe selected due to a high frequency spike (it is difficult to determine with certainty where overall peaks and valleys may occur).

To avoid this problem, an average filter is applied to the histogram to smooth out any high frequency spike. At the same time, the average filter helps lower the level of the most repeated intensities thus resulting in a more level balanced average filtered histogram ($\overline{H}$) where lower frequency trends are more visible and the histogram is smoothed out (see figure 3). Equation 1 is used to apply average filtering to the raw histogram (*H*) where (N_f) is the order of the filter *(N_f >3 and must be odd number)* to be applied, reference index (*i=0, 1, 2, ..., 255*) will be used for the calculation of all resulting ($\overline{H}$) elements and supporting index (*j*) will point to the (*H*) elements needed to be added to the accumulator prior to dividing it by the *(N_f)* number of elements.

$$\bar{H}_i = \frac{1}{N_f} \sum_{j=i-(N_f-1)/2}^{i+(N_f-1)/2} H_j \tag{1}$$

For simplicity, *Hj=0* when (*j*) index is out of bounds (*j<0 or j>255*).

3) We interpret the resulting ($\bar{H}$) to be a discrete one dimensional signal so we can apply a pseudo-derivative ($\dot{H}$) operation (equation 2) to detect maximums and minimums in ($\bar{H}$). The pseudo-derivative differs from the discrete derivative in $\Delta x=N_f -1$ (*Δx=1 for discrete derivative)* and the elements evaluated are equally spaced near left and near right neighboring elements of the discrete signal ($\bar{H}$) (*in discrete derivative, elements are adjacent*). The same value of (N_f) used in equation 1 is used for equation 2. By using the pseudo-derivative small changes sudden changes in ($\bar{H}$) are attenuated whilst considerable changes in ($\bar{H}$) become more appreciable.

$$\dot{H}_i \approx \frac{\bar{H}_{i+(N_f-1)/2} - \bar{H}_{i-(N_f-1)/2}}{N_f - 1} \tag{2}$$

The indexing element (*i = 0, 1, 2, …, 255*) is used in the calculation of all elements of ($\dot{H}$) (see figure 4), also all out of bounds indexed elements of ($\bar{H}$) are considered to be zero. The value of N_f used was 13 which netted good results during experimentation.

4) Finally a routine is used to detect changes in polarity of the slopes starting from $\dot{H}_0$ to $\dot{H}_{255}$. A change from positive to negative is stored as a peak and a change from negative to positive is stored as a valley.

5) The location (or index) of the first considerable valley in the histogram is taken as the intensity threshold [2]. (See resulting effect applying dark intensity threshold in iris image in figure 5).

It is noteworthy to state that no further segmentation will be applied to remove occluding eyelashes, which may still be present after the dark threshold applied to the iris image.

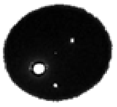


Fig. 5. Coarse pupil localization obtained by applying intensity threshold to iris image.

3 Canny Edge Detection

Within the algorithms available for contour detection in image processing, we use a proven method known as Canny edge detection [4]. Canny edge detection employs a combination of low pass image filtering, followed by high pass filtering to obtain horizontal and vertical gradient component images. The contour image (see figure 6) formed by the magnitude of the component images is further thinned out through a phase/angle analysis.

Once Canny edge detection has been applied to the pre-segmented image (figure 5), the process of collecting pupil/iris contour points can be started. The pixel coordinate collection process used is horizontal row scanning, approaching from the left as well as approaching from the right. The approaching scan from either direction is stopped where any appreciable pixel is detected. Thus, detection of only two or less pixels per horizontal row in the image is possible. Pixels detected have their coordinates stored in a 2D array. Even though it is possible that points belonging to an anomaly such as light reflection may be selected, these will not conform to the constraints of the next process of pupil center localization and as such be subsequently discarded.

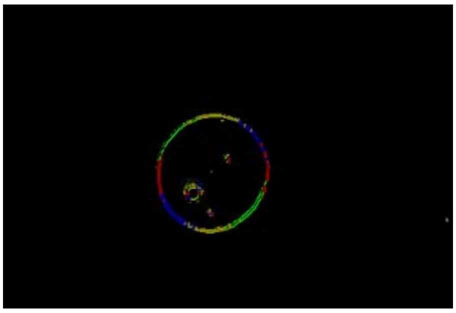


Fig. 6. The resulting image after Canny edge detection.

4 Pupil Center and Pupil/Iris Boundary Localization using Geometrical Methods

The method for pupil center localization parts from the simple geometry problem of formulating a circle equation from three points.

The summarized solution is as follows: Given three points in a 2D plane (figure 7a), midpoints between the original points are calculated (figure 7b). Two new line equations are formulated in such way that they pass through the calculated midpoints and are perpendicular to the previous lines. The intersection point of these new lines is considered to be the circle center (figure 7c). The radius can be obtained from calculating the distance from the circle center to any of the original three points (figure 7d). Finally an appropriate circle equation can be formulated parting from a known circle center and a known radius (figure 7d). The detailed process is explained fairly well by [5].

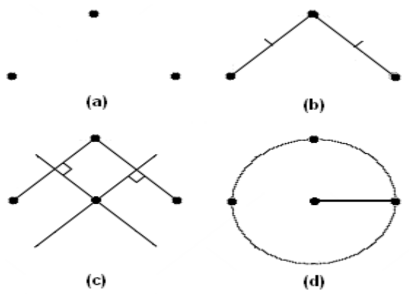


Fig. 7. Formulating circle equation, given 3 points in 2D plane.

While all most any set of three points in a 2D plane can be used to formulate a circle. This is less true for four points, as an implicit constraint arises where these points have to belong to a circular type of object for them to have one conceding intersection point from their derived intersection lines [1]. The constraint is even more severe, when more than four points are taken into account.

The method described in this paper selects four arbitrary points that are evenly distributed, as they should be points that are representative of the remaining contours. These must be selected from the 2D array where pixel coordinates where stored after Canny Edge Detection. Midpoints which contain the necessary data to formulate the intersecting lines [5] are then calculated.

We analyze the slope angle (θ) of the intersection lines prior to formulating them in rectangular format. By utilizing the “arctan2” function, in which computer compilers handle the numerator (m_y) and denominator (m_x) of the slope (m) as separate parameters, division by zero can be avoided (see equation 3). With the use of the “arctan2” function, (θ) will yield values ranging from (–π to π).

$$\theta = arctan2(m_y, m_x) \tag{3}$$

To simplify the interpretation of the slope angle (θ), we add (π) to a negative result (as seen in equation 4).

$$\theta = \begin{cases} \theta + \pi, & \theta < 0 \\ \theta, & \theta \geq 0 \end{cases} \tag{4}$$

Based on the slope angle (θ) (see equation 5) we formulate the linear equations, firstly in the slope intercept form (*y=mx+b for near horizontal lines or x=my+b for near vertical lines*), where (b) is the intercept along the appropriate axis [5]. Then secondly in the standard form ($k_1x+k_2y=k_3$), which is achieved by isolating (b), where (k_1, k_2 *and* k_3) are the known constants.

Once we have formulated at least three intersection lines in standard form (see left side of equation 5) we can proceed to calculate an intersection point for all lines solving the set of linear equations using Kramer determinants.

$$k_3 x + k_1 y = k_2 \leftarrow \begin{cases} y = mx + b, & \theta \leq \frac{\pi}{4} \\ x = my + b, & \frac{\pi}{4} < \theta < \frac{3\pi}{4} \\ y = mx + b, & \theta \leq \frac{3\pi}{4} \end{cases} \qquad (5)$$

In the case where we have only one solution point, for the moment we consider this to be the pupil center and proceed to obtain the pupil/iris radius by calculating the distance to any of four originally selected contour points. Finally we measure the number of pixels in the contour image found along a circle perimeter (with the calculated pupil center and pupil radius) to ascertain the results.

In the case where there is no single solution point for the intersecting lines, four new contour points are selected until a single solution point for the new intersecting lines is established.

5 Iris/Sclera Boundary Localization

As stated previously, this paper assumes that two concentric circles that define the pupil/iris boundary and the iris/sclera boundary are centered on a single point [1]. The iris/sclera boundary will be detected by identifying a reasonable contrast change in data collected from an outwards scan parting from the pupil/iris boundary. In practice, most iris images contain anomalies such as light reflection which can be misinterpreted as a contrast change when scanning for the iris/sclera boundary. Or the regions containing the eyelids or eyelashes may also be misinterpreted as a significant contrast change. For small anomalies located in the iris region, the data collected from problem pixel intensities can be average filtered with like pixels in the iris that share the same radius (or distance) from the pupil center.

5.1 Pixels Scan using Circular Motion

A model for obtaining the pixel indexes with the intention of scanning pixels along a circle perimeter within the image is needed. The model devised is based on the formula for parametric circular motion shown with equation 6.

$$P(k) = (r \cdot \cos \omega \qquad (6)$$

Where (r) is the radius of the circle, (ω) is the angular frequency of the circular motion, (x_c) and (y_c) are the horizontal and vertical displacement of the circle center (for our application the index of the located pupil center) and (k) is the function variable.

Parting from the relationship of angular frequency (ω) and period (T) as shown in equation 7 where (T) is the perimeter of the circle for this application of scanning along the circle perimeter in one cycle, (ω) can be simplified in terms of (r).

$$\omega = \frac{2\pi}{T} = \frac{2\pi}{2\pi r} = \frac{1}{r} \tag{7}$$

Thus equation 6 and equation 7 can be combined for expressing (ω) in terms of (r) for this application as shown in equation 8:

$$\vec{r}(k) = \left(r\cos\frac{k}{r} + x_c, r\sin\frac{k}{r} + y_c \right) \tag{8}$$

5.2 Mean Pixel Intensity of a Circle Perimeter Scan

Given that the pupil center in the image has been located, and that the distance from this point to the pupil/iris boundary is known. This will be our initial radius (r_i) for scanning outwards in a circular motion. After the mean pixel intensity data ($\bar{C}_r$) has been collected and stored for this radius (r_i), the radius will be increased by one pixel and new data will be collected for (r_{i+1}) this data collection process will continue with increasing radii until an established final radius (r_f) is reached or the scan yields an out of bounds index in the image.

The following equation 9, is an example of obtaining the mean pixel intensities ($\bar{C}_r$) in the image (Img) that were selected by indexing in a circular motion. The perimeter comes into play in the calculation of the mean as its value contains the number of pixel intensities that were scanned and as a stopping limit for the parametric variable (k).

$$\bar{C}_r = \frac{1}{2\pi r} \sum_{k=0}^{2\pi r - 1} Img\left[r\cos\left(\frac{k}{r}\right) + x_c, r\sin\left(\frac{k}{r}\right) + y_c \right] \tag{9}$$

In addition, the model must also have the capacity to scan only certain segments of the circle perimeter, delimited by fixed constraint angles with respect to the pupil center. In this manner, a good deal of pixel information regarding eyelashes and eyelids is omitted. The ones proposed in the related work by [1] are used: From -π/4 to π/4 and from 3π/4 to 5π/4 (see figure 8 for a projection of the delimited segments).

These delimiting angles can be translated to a corresponding (k) variable value (being 2πr one complete cycle): From -πr/4 to πr/4 and from 3πr/4 to 5πr/4.

Taking these new segmented circle restrictions and applying them to equation 9 yields the delimited equation 10.

$$\bar{C}_r = \frac{1}{\pi r} \sum_{k=\frac{-\pi r}{4}}^{\frac{\pi r}{4}} Img\left[r\cos\left(\frac{k}{r}\right) + x_c, r\sin\left(\frac{k}{r}\right) + y_c \right] + \frac{1}{\pi r} \sum_{k=\frac{3\pi r}{4}}^{\frac{5\pi r}{4}} Img\left[r\cos\left(\frac{k}{r}\right) + x_c, r\sin\left(\frac{k}{r}\right) + y_c \right] \tag{10}$$

In many iris images, the iris/sclera boundary is not well defined and usually presents a low contrast change. When analyzing the data ($\bar{C}_{r_i}, \bar{C}_{r_i+1}, \dots, \bar{C}_{r_f}$) it is difficult to establish with certainty which radius represents the iris/sclera boundary. For this reason, consecutive circle scans are grouped into layers representing the average pixel intensity conformed of several circle scans. The layer size (L_{size}) value chosen during the experimentation was three (the same as used in [1]), which netted good results for the "NIR still eye image" set [3].

The mean layer intensity ($\bar{L}$) (below in equation 11) can be used to represent a group of consecutive ($\bar{C}_r$) data from a starting radius (r_s) to the ending radius (r_s+ L_{size}-1) within the layer.

$$\bar{L} = \frac{1}{L_{size}} \sum_{r=r_s}^{r_s+L_{size}-1} \bar{C}_r \qquad \textbf{(11)}$$

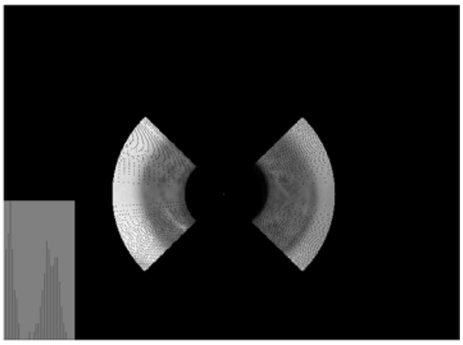


Fig. 8. Outwards scan to find the largest contrast change parting from pupil/iris boundary (lower left plot: mean layer intensity vs. layer radius).

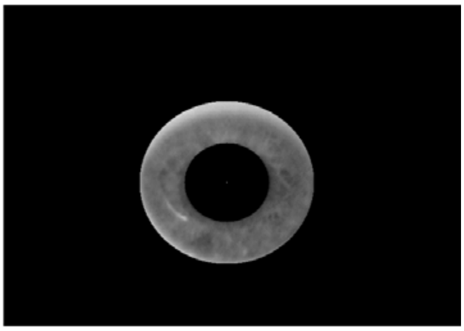


Fig. 9. Segmented iris image of figure 1.

Thus by analyzing the consecutive mean layer intensities ($\bar{L}$), we can find the highest contrast change from one layer to another (the first few layers will be ignored as some of the dark pixels of the pupil may intervene in the first layers, see figure 8).

The layer radius in which the high contrast change occurs will be recorded as the radius that defines the iris/sclera boundary. The final segmented image for figure 1 is seen in figure 9.

6 Results/Segmentation

Results very similar to the ones found by the method proposed by Liu *et al* [1] where obtained. Even in some images with occluded irises, the pupil center point, iris/pupil boundary, and iris/sclera boundary where genuinely located by the method (see figure 10 and figure 11).

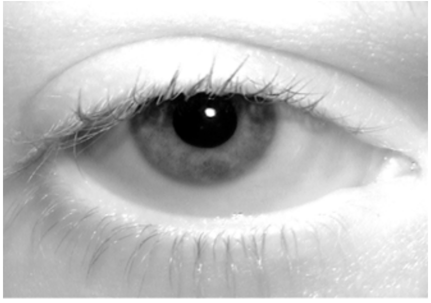


Fig. 10. Original iris image.

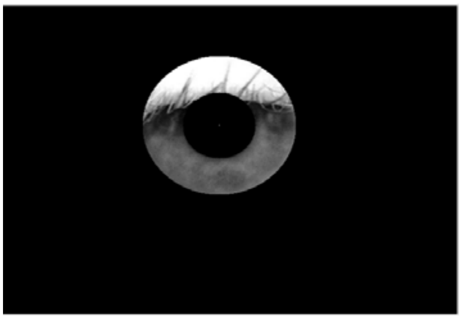


Fig. 11. Segmented iris image.

The geometrical method relies in many ways on the localization of the pupil center. As stated, genuine circle points must be selected to construct appropriate lines for the pupil center point localization. Thus problems may arise when there are too many contours left after canny edge detection (see figure 12, figure 13 and figure 14).

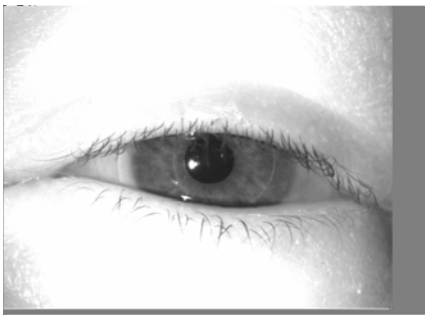


Fig. 12. Original iris image.

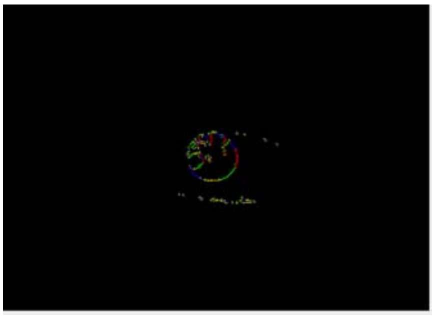


Fig. 13. Eyelash contours remain after Canny edge detection.

This would affect the next stage where non-genuine circle points would be collected during the circle contour point collecting routine (effect seen in figure 14).

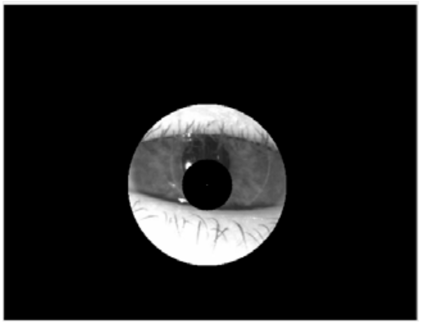


Fig. 14. Failed Iris Segmentation.

7 Conclusions

Parting from the assumption that iris images taken from a controlled iris image set contain considerably well defined Euclidean circles, the method proposed provides good iris localization with less computational latency by reducing image operations and dropping Hough circle detection on the iris image. The method proposed suffers when iris images of very low resolution are used (accuracy problems) and when there are far too many anomalies in the iris image (after canny edge detection).

7.1 Comparison with Previous Methods

Practical comparison with the previous methods mentioned in this article is not easily achieved. For comparison with the method proposed by Liu *et al* [1], while the results are visually similar for iris segmentation, in our proposed method no eye image acquisition system is being used (fundamental component of method [1]). As for comparison with the method proposed by Sun *et al* [2], while we share the similar component of histogram analysis, the focus of our method was always intended to reduce computational demanding operations like image morphology or Hough circle detection present in method [2]. Furthermore ascertaining a quantifiable measurement between the three methods for comparison is difficult, since there is no standardized measurement for iris segmentation.

It is also to be noted that state of the art methods focus on the solution to the iris biometric identification problem as a whole; however, iris segmentation does not conform the total solution in itself, but as a component can still be very useful.

Acknowledgments. This work was supported by IPN-SIP20100030.

References

1. Liu, Y., Yuan, S., Zhu, X., Cu, Q.: A practical Iris Acquisition System and A Fast Edges Locating Algorithm in Iris Recognition. In: IMTC 2003-Instrumentation and Measurement Technology Conference, Vail, CO, USA, pp. 20-22 (2003)
2. Sun, C., Zhou, C., Liang, Y., Liu X.: Study and Improvement of Iris Location Algorithm. In: ICB 2006, LNCS 2832, pp.436-442 (2005)
3. NIST Multiple Biometric Grand Challenge. http://face.nist.gov/mbgc/
4. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 8 (6):679–698 (1986)
5. Bourke, P.: Equation of a circle from 3 points (2 dimensions), http://local.wasp.uwa.edu.au/~pbourke/geometry/circlefrom3/

A Support Vector Machine Based Approach to Evaluation of the Quality of Patterns

Luis Horna[1], Ricardo Barrón[2], and Salvador Godoy[3]

Centro de Inevstigación en Computación , Instituto Politécnico Nacional,
Laboratorio de Inteligencia Artificial,
Av. Juan de Dios Bátiz s/n, México, D.F., 07738, Mexico
[1]chornab08@sagitario.ipn.mx
[2]rbarron@cic.ipn.mx
[3]sgodoyc@cic.ipn.mx

Abstract. In the routine task of Pattern Recognition, time and effort is invested to extract characteristics, and create large pattern data sets that do not always behave as expected when training a pattern recognition system. In those cases it is worth to try to evaluate if the patterns that have been extracted are either of high or low quality. In this paper, we propose a method to evaluate the quality of patterns by using support vector machines.

Key words: Support Vector Machine, Quality of Patterns.

1 Introduction

One of the problems that is always present in pattern recognition is to know whether patterns from different classes are linearly separable, which can be seen as a measure of quality of the features being used. This problem can be evaluated by making scatter plots and visually evaluate, computing correlation[10] or distance between classes [4], evaluating the taxonomy of the classes, as well as making a principal component analysis (PCA) [3], [11].

When dealing with pattern recognition tasks it is common to find classes or clusters with very little separation, intersection near their borders or, in the worst case, completely overlapping Fig.(1). In such situations, it is desirable to identify which classes have this problems as well as which patterns are causing this problem. Evaluating the quality of the patterns being used is essential in order to select a classifier suitable for a given situation, if the quality of data is poor one could end up using classifiers whose computational cost is high so as to compensate. Measuring the quality of data can also be helpful to determine whether to change the features being used.

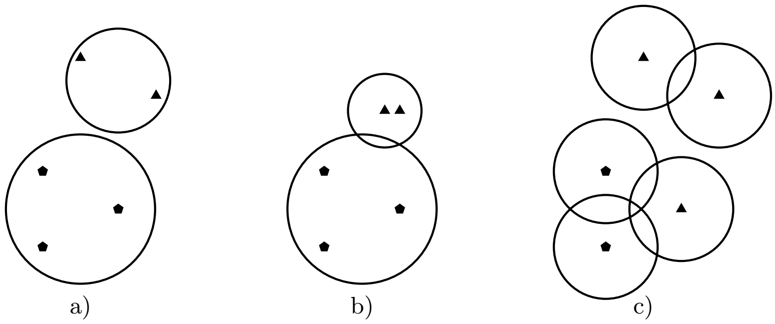


Fig. 1: a) Classes or clusters with very little separation, b) Intersection near their borders, c) Classes completely overlapping.

In this paper we propose a novel index to assess the quality of patterns by identifying three different situations:

1.- ratio of patterns from each class that could be correctly separated
2.- ratio of patterns from each class that could not be correctly separated.
3.- ratio of patterns that are classified as elements of other classes.

desirable characteristics of high-quality patterns would be to minimise this three different possibilities. In this paper we will develop a novel methodology to evaluate this three possibilities, our methodology will make use of SVM for such purpose.

2 Proposed Methodology

Assessing the quality of patterns in large data-sets is, generally, not an easy task. We have selected support vector machines (SVMs) over other classifiers, because their capability of handling large data sets [2], [7], [9],[12], as well as handling vectors of large numbers of features [8] , SVMs can deal with non-linearly separable classes , and SVMs are widely used for data mining [5], [13] which is similar to our problem.

From a formal point of view a perfectly linearly separable set of classes would be one that satisfies:

$$C_1 \cap C_2 \cap ... \cap C_n = \emptyset \tag{1}$$

where C_i is a class, now in reality

$$C_1 \cap C_2 \cap ... \cap C_n = W \tag{2}$$

the cardinality of W depends on several factors, but in general the better the quality of patterns and classifier the smallest W is.

The proposed solution in this paper makes the assumption that if a pattern is not classified correctly during training then it must be removed, under common training approaches this would be a mistake. However, since the goal in this particular problem is precisely trying to find those patterns and in consequence the classes that have problems of linear separability is ideal to remove them.

Our approach consists in two phases. First, for a fundamental set F we create a support vector machine (we use a C-SVC) for each of the C_i classes (i.e dividing F into C_i and non-C_i), then evaluate its performance and remove those patterns that were not correctly classified until performance is satisfactory, create Q_i with the reduced set C_i. Second, create a set S_i with all the patterns that could not be correctly classified when creating Q_i.

It is clear that $Q_i \cap S_i$ is not necessarily $\emptyset$, this follows from the fact that some $X_k \subset C_i$ may have ended up in S_i. It should also be pointed out that this process must have a finishing condition other than reaching some performance, because there is no warranty it will ever be achieved, instead a maximum number of iterations must be used. Once all Q_i and S_i have been computed, all that rests it to evaluate the quality of the patterns. It should be pointed out that the SVMs used have to use the same kernel function, no particular restriction is imposed when selecting the kernel function.

In order to measure the quality of patterns, we measure:

1.- ratio of patterns from each class that could be correctly separated, denoted by ψ.
2.- ratio of patterns from each class that could not be correctly separated,denoted by χ
3.- ratio of patterns that are classified as elements of other classes, denoted by φ.

each of the ratios is respectively computed by:

$$\psi_i = \frac{card(Q_i)}{card(C_i)} \tag{3}$$

where values close to zero represent very low pattern quality, and values close to one represent high quality patterns.

$$\chi_i = \frac{card(\overline{Q_i \cap C_i})}{card(C_i)} \tag{4}$$

χ_i represents exactly the opposite of ψ_i.

$$\varphi_i = \frac{card(S_i)}{card(F)} \tag{5}$$

finally, if φ_i is very close to zero it means that very few patterns are causing problems when classifying C_i, and values very close to one are a sign that almost every pattern causes problems.

After computing ψ_i, χ_i, and φ_i, it is clear that Q_i and S_i contain only the patterns that could be classified with the highest accuracy, this could be used to assign weights to each class C_i, and Q_i could be used as a model to create auto-associative memories that work as a filter.

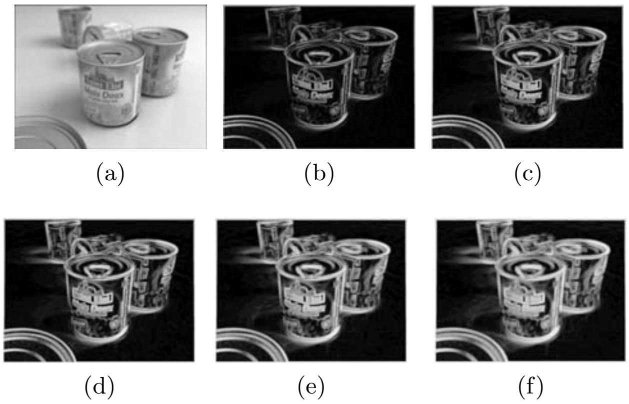
At this point two situations must be clarified, first our approach uses the SVMs to detect which patterns are causing problems when training, on eahch iteration we only keep those that are useful. Second, the proposed indexes aim to measure the quality of patterns of a given data set taking into account the patterns that cause problems during training.

3 Experimental Results

In this section, we present two experiments, one to evaluate poor quality patterns another with high-quality. The first experiment was an attempt to create pattern recognition system that could establish relationship between an image I and a depth map D. The second experiment consist in evaluating the quality of well known data sets.

In the first experiment, the features extracted are the standard deviations of the image using different scales of 3x3,5x5,7x7,9x9, and 11x11, which are used to create a feature vector of dimension 5. Image I is sampled only along the edges, and the fundamental set has the form:

$F = \{((L_{3x3}, L_{5x5}, L_{7x7}, L_{9x9}, L_{11x11}), d_1), ...((L_{3x3}, L_{5x5}, L_{7x7}, L_{9x9}, L_{11x11})), d_n)$


(a) (b) (c)

(d) (e) (f)

Fig. 2: Images representing the characteristic extracted.

where $d_i \in [0, 255]$, Fig.(2) shows an image and each of the features measured. In order to make this experiment easy to understand, depths are grouped in intervals of five, for instance C_1 would group all d_i in $[0 - 4]$. The SVMs use a radial basis kernel with parameters $\gamma = 0.20$ and $C = 1$.

From an empirical point of view it may seem that the features from Fig.(3) should provide information to accomplish the goal. However, when applying

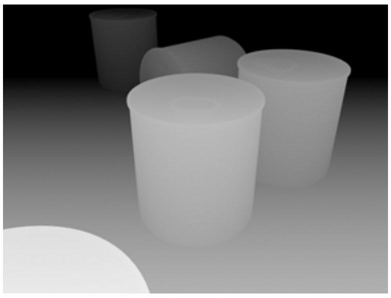


Fig. 3: Depth map used as target.

our methodology the following information about the patterns being used is obtained, tab.(1) shows the results for five classes.

Table 1: Result of the first experiment.

class	ψ	χ	φ
0-4	0.72	0.28	0.01
89-93	0.0	1.0	0.17
224-228	0.16	0.84	0.04

results show that the features are of poor quality and further more classes are completely mixed, which makes data completely useless. This experience shows that it is worth to evaluate the quality of patterns before investing time and effort training and testing different classifiers. The second experiment consists in evaluating the quality of patterns from the well known Iris plant (three classes) and Wine (three classes)[18] data sets, results of their evaluations are shown in tab.(2), tab.(3) and tab.(4), parameters used in the SVM are $\gamma = 1/N$ and $C = 1$,with N =dimension of feature vector.

Table 2: ψ values

Data set	ψ_1	ψ_2	ψ_3	ψ_4
Iris plant	1	0.91	1.0	n/a
Wine	0.93	0.97	1.0	n/a

Table 3: χ values

Data set	χ_1	χ_2	χ_3	χ_4
Iris plant	0.0	0.09	0	n/a
Wine	0.07	0.03	0.0	n/a

Table 4: φ values

Data set	φ_1	φ_2	φ_3	φ_4
Iris plant	0.0	0.05	0.05	n/a
Wine	0.02	0.01	0.01	n/a

From these results it can be confirmed that the quality of this well known training sets is good, note that the classes that have small value of ψ_i are those classes that are mixed in some degree with other classes. Which is expected as is commonly reported in literature.

In this section we have presented how our methodology works on both high-quality patterns and completely useless patterns, we have also shown the capability of our approach to detect classes that are overlapped, in all experiments libsvm [1] was used.

4 Future Work

It remains to evaluate our method on other patterns recognition approaches such as neural networks or distance based classifiers. Another interesting aspect to develop in future research is the computing of weights for classes using our methodology. It also would be interesting to investigate the use of Q_i to create auto-associative memories, which could be used to improve the accuracy of classification when using a given classifier. Finally it remains to expand our methodology to detect relevant/irrelevant features from patterns.

5 Conclusion

In this paper a novel method for evaluating the quality of patterns has been presented, also it has been shown how our method can be used to create sets of "model" patterns, which could be later used to filter unknown patterns. We believe that the proposed methodology could be an alternative to exiting techniques for evaluating the quality of patterns. From our perspective the proposed methodology is useful because it extracts the set of patterns causes most of the problems when training, which could be analized to either change the features used or label such set as new class.

References

1. C-C.Chang, C-J.Lin: LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/ cjlin/libsvm/ (2001)
2. Y.L. Murphey, Chen Zhihang, M. Putrus, L. Feldkamp: SVM learning from large training data set. In: Proceedings of the International Joint Conference on Neural Networks, Vol.4, pp. 2860 - 2865 (2003)
3. A.V. Anghelescu, I.B. Muchnik: Combinatorial PCA and SVM methods for feature selection in learning classifications (applications to text categorisation), In: KIMAS '03, pp. 491-496 (2003)
4. I. W. Tsang, J. T. Kwok, P.-M. Cheung: Core vector machines: fast SVM training on very large data sets.Journal of Machine Learning Research, 6, pp. 363-392 (2005)
5. L. Yu, S. Wang, K. K. Lai: Mining Stock Market Tendency Using GA-Based Support Vector Machines, Internet and Network Economics. Vol. 3828/2005, Berlin, Springer, pp. 336-345 (2005)
6. J. Cervantes, X. Li, W. Yu , J. Bejarano: Multi-Class Support Vector Machines for Large Data Sets via Minimum Enclosing Ball Clustering. In : CEEE 2007, Mexico (2007)

7. J. Cervantes, X. Li, W. Yu , J. Bejarano: Two-stage svm classification for large data sets via randomly reducing and recovering training data. In: IEEE International Conference on Systems, Man and Cybernetics, ISIC., pp. 3633 - 3638 , Montreal, Canada (2007)
8. Henryk Maciejewski: Quality of Feature Selection Based on Microarray Gene Expression Data. Computational Science – ICCS 2008, Vol. 5103/2008, pp. 140-147, Berlin, Springer (2008)
9. Zhi-Qiang Zeng, Hua-Rong Xu, Yan-Qi Xie, Ji Gao: A geometric approach to train SVM on very large data sets. In: ISKE 2008., Vol.1, pp. 991 - 996, Xiamen, China (2008)
10. Jirong Li: Feature Selection Based on Correlation between Fuzzy Features and Optimal Fuzzy-Valued Feature Subset Selection, In: IIHMSP '08, pp. 775 - 778, Harbin, China (2008)
11. Yihui Luo, Shuchu Xiong, Sichun Wang: A PCA Based Unsupervised Feature Selection Algorithm. In: WGEC '08, pp. 299 - 302, Hubei, China (2008)
12. J. D. Wang, H. C. Liu: Evaluating the Ambiguities between Two Classes via Euclidean Distance. Asian Journal of Health and Information Sciences, Vol. 4, No. 1, Taiwan, pp. 21-35, (2009)
13. P.C, C.Y. Tsai, C.H Huang, C.Y. Fan: Application of a Case Base Reasoning Based Support Vector Machine for Financial Time Series Data Forecasting, Emerging Intelligent Computing Technology and Applications, Vol. 5755/2009 ,pp. 294-304, Berlin, Springer (2009)
14. E.Y. Cheu, C. K. Kwoh, Z. Zhou: On the Two-level Hybrid Clustering Algorithm, Nanyang Technological University, Singapore.
15. C. van der Walt, E. Barnard: Data characteristics that determine classifier performance. Human Language Technologies Research Group Meraka Institute, Pretoria: South Africa.
16. C. Domeniconi, D. Gunopulos: Efficient Local Flexible Nearest Neighbor Classification, Dept. of Computer Science, University of California, Riverside: USA.
17. L. Zhang, F. Lin, B. Zhang: Support vector machine learning for image retrival. State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University, Beijing: China.
18. UCI machine learning repository. http://archive.ics.uci.edu/ml/.

Robotics

Coordinated Object Manipulation by a Group of Robots

Roberto Hernandez[1], Rafael Castro[2], and Humberto Sossa[1]

[1] Department of Artificial Intelligence, CIC-IPN, Mexico D.F., Mexico
[2] Department of Electrical Engineering, CINVESTAV-IPN, Mexico D.F., Mexico
rhar.ha@gmail.com, rcastro@cinvestav.mx, hsossa@cic.ipn.mx

Abstract. A control strategy for the control and coordination of two robots for the manipulation of objects is presented in this work. The ability to maintain a prescribed formation allows the robots to manipulate rigid objects, guaranteeing that the desired task is accomplished. The control goal is that the follower robots follow the leading robot with an unknown motion in the working area. For image analysis, an acquisition camera was mounted at the top of the working area. One of the primary advantages of this strategy is the facility to measure the position and orientation of all the robots, objects and obstacles. The vision system makes use of image analysis and pattern recognition techniques to allow the group of robots to accomplish their task. Preliminary experimental results performed with two Khepera II mobile robots show the effectiveness of the proposal.

Keywords: Mobile Robots, Artificial Vision, Pattern Recognition, Rigid Objects, Coordination of Robots.

1 Introduction

In order that a group of mobile robots follow a set of paths in working area to perform a task, it is necessary to know their trajectories by measuring their individual positions and orientations. For this, an image analysis and control system is needed.

The task of transporting objects by cooperative communities of robots includes, among several issues: avoiding collisions, calculating the trajectories of the robots, as well as the positions and orientations of the objects to be manipulated. Most robots must change not only their positions but also their directions to move the object through the work area, given that we have a fixed distance among the robots while transporting the object.

The usage of a community of robots allows efficiently manipulate objects better than with a single robot. Also the use of centralization information allows a total control of all the operations, processes and vision tasks.

In this work we propose a novel vision based-control law that, as we will see, allows manipulating of objects using a community of robots. Without a loss of generality, in this research we work with two robots only, one leader robot and one follower robot.

The rest of the paper is organized as follows. In section two we describe some of the most important works related with the present investigation. In section three we describe the proposed methodology, while in section four we present some initial experiments where our methodology is tested. Finally, in section five, we give some conclusions and directions for further research.

2 State of the Art

When we explore the literature we find either works related with the manipulation of objects by a single robot or works related with control of a formation of robots. Until now, to our knowledge, we had not found reported works discussing the problem of cooperative manipulation of objects by means of a formation control law. The only works that we have found in the literature that discuss the problem of cooperative object manipulation make use of heuristics [1] or centralized control [2].

In [1] the authors describe a methodology that allows a group of three robots to form a triangular formation. The leading robot has a set of marks in his posterior part that allow the follower robot to follow the leading robot. The follower robots have an acquisition camera mounted on them to perform the necessary image analysis to accomplish the task. Communication among the three is complicated, and most of the times delays among the robots is present, provoking that the triangular formation is lost.

In [2] a controlling computer makes use of a fixed camera mounted over the working area and odometry to determine the initial position and orientations of the robots and the objects to be manipulated. The computer then sends commands to the robots so they can follow their trajectories to reach a given object. In this case the control is an open loop. In the experiments presented by the authors, the robots reach their desired positions but only in an approximate way. The goal to manipulate a given is never accomplished as reported in the paper.

Other investigations have been also reported in the works [3] and [4], but they not deal with the accomplishment of the task of cooperative object manipulation, either because image analysis fails or because the desired robots are lost. A lot of work has to be performed in this direction.

In [5], robots calculate an actual relationship with a neighbor using only sensor readings. The robot communicates locally with discrepancies in its desired and actual relationships to neighboring cells. Correcting for these discrepancies produces robot movements that result the overall organization of the desired global structure. The achieved formations of the robots using sensors was extended by a robot that uses a video camera, but this also fails to make them work with real robots or manipulating objects.

In [6] behavior-producing modules are used: a robot is capable of following other while avoiding obstacles. Since no absolute referencing system is used, robots have to rotate themselves to detect the presence of the other robot nearby, the robots used the color camera according to the position of the colored-blob in the image to keep-formation leader. In this work, the qualifications obtained cannot be used to manipulate objects because they make triangular shapes or online.

2.1 Contributions

Given the published literature, this study provides some contributions that have not been done before and provided some modifications:

– Presents an alternative to manipulate objects using a visual control in a leader follower scheme.
– Is able to solve a small control errors caused by system dynamics.
– Generate of trajectories that can provide one of the minimal paths that may be able to avoid collisions.

3 Methodology

In this paper, the problem of cooperative object manipulation is divided into two stages, as follows. The first stage is composed of five parts:

1. Identification of the position and orientation of all items in the working area (robots, objects to be manipulated and barriers).
2. Determination of the shape of an object to be manipulated.
3. Determination of the number of robots needed to manipulate this object.
4. Determination of the path for each robot to reach their desired final positions.
5. Control of the robots to follow the computed trajectories to reach final manipulation positions.

Second stage is also composed of five parts, as follows:

1. Control of the robots to take the object.
2. Computation of the path of the leading robot to move in a cooperative way with the follower robots the object to a destination position.
3. Control the leader robot to follow the computed path.
4. Use of formation control low to move the whole set of robots to transport the object.
5. Put the object on the final position.

In the next sub-sections, we describe each if these steps in more detail.

3.1 Formation Control Model

In this work, a formation control based on the so called leader-follower scheme is considered [7]. Also, the controller for this formation is designed using passivity and sliding mode techniques in accordance to [8]. The formation control stage allows the movement of the follower robot to a given distance and angle with respect to the leader robot. The way in which these signals change is set in the stage of trajectory generation. The leader-follower formation considered is shown in Fig. 1, where $X-Y$are the fixed coordinates and x$-y$ are the Cartesian coordinates fixed in the body of the leader robot. (X_L, Y_L) and (X_F, Y_F) denote the global positions of the leader and follower, respectively whitv_L and $v_F$are the linear velocities of the leader and follower; θ_L and $\theta_F$are the respective angles of orientation; l is the relative distance between the leader and follower while $\varphi$is the orientation of the follower with respect to the leader.

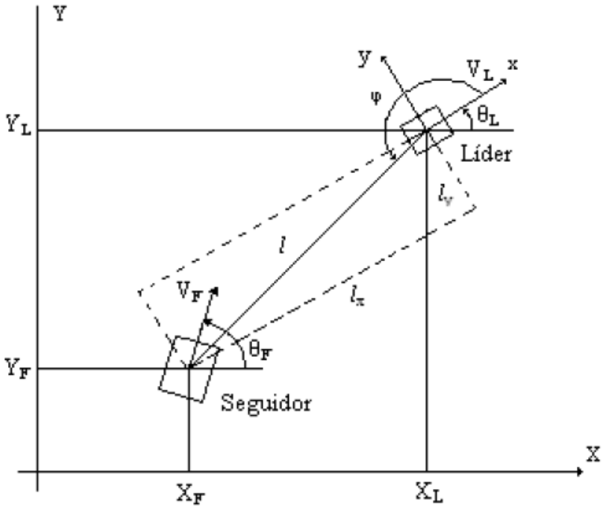


Fig. 1. Leader – follower formation.

The leader and follower kinematic models are then described by [9]

$$\dot{X}_L = V_L \cos\theta_L$$

$$\dot{Y}_L = V_L \sin\theta_L$$

$$\dot{\theta}_L = \omega_L \ . \qquad (1)$$

$$\dot{X}_F = V_F \cos\theta_F$$

$$\dot{Y}_F = V_F \sin\theta_F$$

$$\dot{\theta}_F = \omega_F \ . \qquad (2)$$

where ω_L and ω_F are the leader's and follower's angular velocities, respectively. Projecting the relative distance between the leader and follower along the x-y Cartesian coordinates leads to the expressions

$$l_x = -(X_L - X_F)\cos\theta_L - (Y_L - Y_F)\sin\theta_L$$

$$l_y = \ (X_L - X_F)\sin\theta_L - (Y_L - Y_F)\cos\theta_L \ . \qquad (3)$$

where $l_x$and l_y are the follower's relative positions along the x and y directions, respectively. In order to maintain a desired formation, it is needed that $l_x \rightarrow l_x{}^d$ and $l_y \rightarrow l_y{}^d$, where $l_x{}^d$ and$l_y{}^d$ are the desired relative position along the x and y directions, respectively.

Here, it is required that the desired relative distance between the leader and follower, l^d to be constant in order to be manipulate a class of rigid objects. On the other hand, to move the object requires that the desired angle φ^d to be time varying in accordance to a desired path. The following error signals are defined:

$$e_x = l_x^d - l_x, e_y = l_y^d - l_y, \; e_\theta = \theta_F - \theta_L \,. \qquad \textbf{(4)}$$

Using the kinematic model (1), (2), the projections given by (3) together with some trigonometric identities and the so called *nonholonomic restrictions*, the dynamics of the error signals take the form

$$\dot{e}_x = \; e_y\omega_L - v_F\cos e_\theta + f_1 + \Delta f_1$$

$$\dot{e}_y = -e_x\omega_L - v_F\sin e_\theta + f_2 + \Delta f_2$$

$$\dot{e}_\theta = \omega_F - \omega_L \,. \qquad \textbf{(5)}$$

where

$$f_1 = \; - l_0\dot{\varphi}^d\sin\varphi^d - \omega_L l_0\sin\varphi^d + v_L$$

$$f_2 = \; l_0\dot{\varphi}^d\cos\varphi^d + \omega_L l_0\cos\varphi^d \,. \qquad \textbf{(6)}$$

with l^d = l_0 = constant$\Delta f_1$and Δf_2 represent uncertain terms due to bad measurements or noise in the sensors and are bounded. It is important to note that $f_1$and f_2 are known bounded functions since it is assumed that φ^d, $\omega_L$and v_L are also smooth and bounded.

The control strategy for $\omega_F$and v_F that would achieve the objective of manipulating some kind of rigid objects, even in the presence of the uncertain terms $\Delta f_1$and Δf_2 is based on the results presented in [8] and [9]. This strategy consists of a control action when $\omega_L \neq 0$(the leading robot turns around itself) and when $\omega_L = 0$ (the leader robot does not rotate around itself) and are briefly described in that follows.

3.1.1 Case when $\omega_L \neq 0$

In this case a control scheme based on backstepping, passivity equivalence and sliding mode techniques was used. This control is given by the controller

$$\begin{aligned} v_F &= \alpha(e_y, y, t) - \omega_L s_1 - \omega - \gamma_2 sign(s_2) \\ \omega_F &= \omega_L - k_2 e_\theta \; . \end{aligned} \qquad \textbf{(7)}$$

where

$$\alpha(e_y, y, t) = \omega_L e_y + f_1$$

$$-\frac{d}{dx}(\omega_L{}^{-1})[f_2 + k_1 e_y + \gamma_1 sign(s_1)] - (\omega_L{}^{-1})f_2$$

$$-(\omega_L{}^{-1})k_1[-\omega_L y - k_1 a_y + \gamma_1 sign(s_1)]$$

$$-(\omega_L{}^{-1})\gamma_1 \frac{d}{dx} sign(s_1)[-\omega_L y - \gamma_1 sign(s_1)]$$

$$s_1 = e_y + k_1 \int_0^t e_y(\tau) d_\tau$$

$$s_2 = y - \int_0^t v(\tau) d_\tau$$

$$v = \omega_L s_1 - k_2 y$$

$$y = e_x - \alpha(e_y) \; . \qquad \textbf{(8)}$$

with

$$\alpha(s_y) = (\omega_L{}^{-1})[f_2 + k_1 e_y + \gamma_1 sign(s_1)] \; . \qquad \textbf{(9)}$$

γ_1, γ_2, k_1, k_2, and k_3 are positive real constants while the $sign(.)$ function defined in the usual way and an approximation was used for this function so that the time derivative of $\frac{d}{dx} sign(s_1)$ in (8) could be evaluated.

3.1.2 Case when $\omega_L = 0$

The control action given by (7), (8) and (9) can only be used when $\omega_L \neq 0$. However, many trajectories imply the leader's orientation to be constant (for example, when the leader robot moves in a straight line). In this case the control strategy proposed in [7] was used and takes the form

$$v_F = c_1(\tanh e_x \cos e_\theta - \tanh e_y \sin e_\theta) + \xi$$

$$\omega_F = -c_2 e_\theta - c_3 sign(e_\theta) \tanh^2 e_\theta \, . \qquad \textbf{(10)}$$

where c_1, c_2 and c_3 are real positive constants with $c_3 > c_1$ and

$$\xi = \frac{\bar{f}_1 \tanh e_x + \bar{f}_2 \tanh e_y}{\delta + \pi(e_x, e_y, e_\theta)} \, . \qquad \textbf{(11)}$$

with

$$\bar{f}_1 = -l_0 \dot{\varphi}^d \sin \varphi^d + v_L$$

$$\bar{f}_2 = \quad l_0 \dot{\varphi}^d \cos \varphi^d$$

$$\pi(e_x, e_y, e_\theta) = \tanh e_x \cos e_\theta + \tanh e_y \sin e_\theta$$

$$\delta = \begin{cases} 0, & \pi(e_x, e_y, e_\theta) \neq 0 \\ \delta_0, & \pi(e_x, e_y, e_\theta) = 0 \end{cases} \, . \qquad \textbf{(12)}$$

where δ_0 is an arbitrary small real positive constant.

It is important to notice that when integrating the control action (7), (8) and (9) together with the control action (10), (11) and (12) in the realization of a path for the manipulation of the rigid objects considered here a switching between the two control schemes should be considered.

3.2 Vision Control

The image analysis module makes use of a video camera placed at the top of the working area to identify the objects (robots, object to be manipulated and barriers) and to determine their positions and orientations. The camera is placed at a fixed height. We assume that objects are isolated from each other, and can appear in any stable position and orientation [10]. Classification of objects is obtained by means of a linear classifier (a distance classifier) which compares the describing vectors of the objects in the working area with the models stoked in the computer. The describing features used in this work are the compactness factor and first two Hu invariants to translations, rotations and scale changes [11]. First and second order moments were used to find the centers and orientations of the robots and objects in the working area. This information is passed to the generation and control modules to perform their tasks.

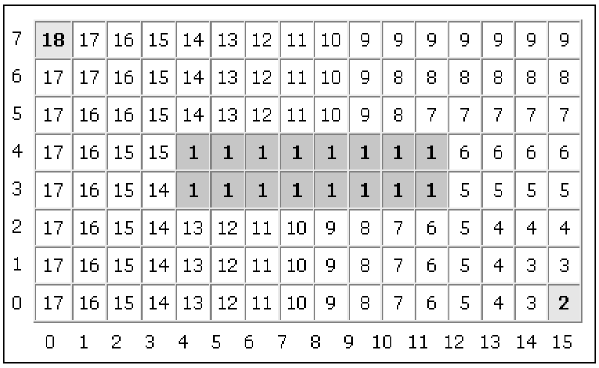
The vision module is responsible for monitoring the position of the robots that perform a task, using the camera fixed, specific data are collected robots and compared with those obtained from the trajectory generation process which represent the path to be followed by the robot at some point toward the state goal, whether it is the object or purpose final task in the event of a mistake this will be corrected by the control stage.

3.3 Trajectory Generation

In order to build a trajectory for the manipulation of the object, the working area is divided into cells in according to the size of the Khepera robot. Such a strategy leads to a path with minimum distances around each cell; this gives computational simplicity as proposed out in [12], the distances are then discretized using an appropriate metric based on the eight neighbors' connectivity of each cell and a label representative tag is assigned to each element of the working area.

Beginning at the point of destination, all the cells associated with the current value (in this case is two) plus one are update in the sense that it the cell does not have pre-assigned label, the updating is carried out with the remaining cells so that the source is reached. This guarantees that the trajectory does not pass thought a cell which is not achievable, such as obstacle.

To generate a path for the leader robot, one considers that moves from an origin cell to the next cell which has a lower numeric value; each cell value represents the distance that separates it from the point of destination. Fig. 2 shows the working area simulated in a computer describing the produce to generate a path from on origin cell. One may notice that the produce described generates paths of maximum longitude.



7	**18**	17	16	15	14	13	12	11	10	9	9	9	9	9	9	9
6	17	17	16	15	14	13	12	11	10	9	8	8	8	8	8	8
5	17	16	16	15	14	13	12	11	10	9	8	7	7	7	7	7
4	17	16	15	15	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	6	6	6	6
3	17	16	15	14	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	5	5	5	5
2	17	16	15	14	13	12	11	10	9	8	7	6	5	4	4	4
1	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	3
0	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	**2**
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Fig. 2. This shows the methodology of the path planning. The final position is the mayor number (*left–top*) and the robot position has number two (right-down), the obstacles are presents with number 1, while the other numbers could be found on the trajectory with minor path length.

The destination point of the first sub-stage is the position where the robots are to be located to manipulate the object. To calculate this, the relative position of the object seen from the video camera is used, drawing a circle of radius 2.3 inches from the center of each cylinder, establishing an ellipse whose focus are the angles between the axis and the centers of each cylinder. The crossing point between the focus and the ellipse is the final destination of each robot and its orientation is perpendicular to the axis.

The selection of the number of robots used in a task depends on the form that has the object to be manipulated, identifies the shape of the object through the chambers allowing video characteristic features of the object to gather and compare them with previously stored in a PC for this selection.

After knowing the number of robots involved in the task should create a path forward from the starting point (current position) to the final position using the method described above, this for each robot.

The manipulation of the object is coordinated by the PC which then sends the appropriate commands to the robots grasp and lift the object enough to be moved, this is achieved with a communication interface based on wireless sync, allowing parallel processing simultaneous sending commands [13].

3.4 Shape of Objects and Handling

There are two variants of objects:

1. The first consists of a cylinder that can be manipulated by a single robot.
2. The second is a rigid object having a structure similar to the table, which must be manipulated by two robots.

In the case of the second object is considerate of two robots for manipulation because the weight is greater than that which supports the robot with its gripper, assuming that the robot can load it as if it were a long tractor-trailer the subject difficult step towards your destination along the trajectory and potential obstacles that can be retrieved to find in their path.

The manipulation of objects is limited to pattern recognition supplied then through the stage of vision, to identify the type of object can determine how many robots are needed for the task of moving the object. But the crucial point of the manipulation is to detect the exact point at which to place each robot to manipulate the object, for this procedure are several factors involved such as:

1. The position and orientation of the object and the robots involved.
2. The final position where the object is placed.
3. The calculation of the final position of the robot path to the object 8que along this work will be appointed as the endpoint).

In the case of the cylinder, the end point is calculated according to the box with the least minimum distance but that is on the opposite side of the final position, while for the second object, it is necessary to make the calculation from end and considerate of the length of the gripper and the two robots must go hand in hand with the same orientation preference on the opposite side of the final position.

Once the robot is placed in the proper position to hold the object proceeds to send the appropriate commands for this not hold, the robot must close your tweezers and lift the object at a distance where the object does not hit the floor when is transported to its destination, nor should be raised that way too because it would cause friction and effort of the servo motors of the robots, 0.5 in was considered sufficient for handling.

4 Results

In this section we present some preliminary results that allow to verifying the effectiveness of the proposal. For this the tests were conducted in a virtual environment. Each stage was tested separately to ensure proper operation before being implemented in the real environment.

To model the physical properties of Khepera II robot and to implement the control low presented in section 3, simulation software was used [14]. In all cases, good results were obtained.

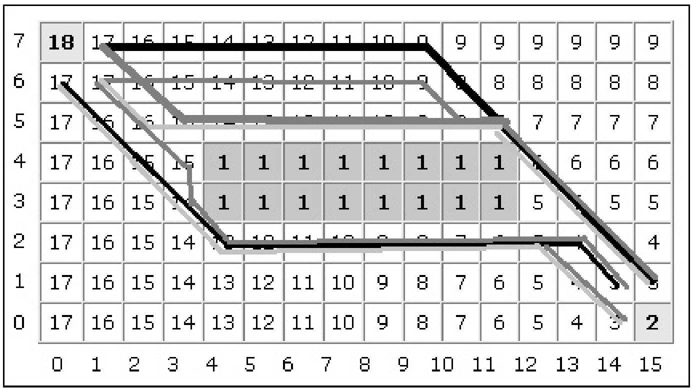
One of the found drawbacks was the initialization of the parameters of the robots. A correct parameter initialization is of vital importance. If they are not correctly initialized huge variations in the computed velocities of the wheels may appear. This might cause that these velocities cannot be implemented with real robots due to their physical constraints.

A solution found to this problem was to pose (position and orientation) the follower robot as close as possible to the virtual path it has to follow.

Path generation was tested on a virtual environment generated by a PC that models the real working area, by generating random conditions for the two robots and the objects to be manipulated. In this case, also good testing results were obtained.

Identification of the elements in the working area was successfully implemented by means of the same programming language used in [15]. The system was tested with 40 images of each item. A correct classification of all the objects was obtained in all cases by comparing the describing features of each element in the working area against their computed models. Also the positions and the orientations of the objects are computed.

In the Fig. 3 we show an example where the possibilities are calculated to move the robot to the final position, to find the minimum distance path that leads to the object enough to count the number of frames for which can happen, for it presents a comparative table showing the optimal path, The figure shows that for this example there are fifteen potential path that may follow, but in any case is the shortest distance and avoid all motor vehicle crash with an obstacle.



7	18	17	16	15	14	13	12	11	10	9	9	9	9	9	9	9
6	17	17	16	15	14	13	12	11	10	9	[illegible]	8	8	8	8	8
5	17	16	16	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	7	7	7	7
4	17	16	15	15	1	1	1	1	1	1	1	1	[illegible]	6	6	6
3	17	16	15	[illegible]	1	1	1	1	1	1	1	1	5	[illegible]	5	5
2	17	16	15	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4
1	17	16	15	14	13	12	11	10	9	8	7	6	5	[illegible]	3	[illegible]
0	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Fig. 3. The figure shows the possible paths solutions found after using the methods for finding the optimal path.

Because in all cases the same number of cells was obtained it is not necessary to present a comparative table of results, where travelling on paths yields the same time, because the cells are all of the same size.

5 Conclusion and Future Work

The paper is a compilation of different techniques and methodologies and others modified some proposals to present an alternative to the manipulation of objects using mobile robots.

The described path generation allows you found the path of least distance to the end point or end state, but not guaranteed to be the only solution will be sufficient for the work presented here. The control used allows follow leading a parallel robot which is suitable for handling objects as they can be transported in a way that is not a straight line.

This work shows the final results though, continues to change and some future changes are bluetooth incorporate a serial connection to replace the busy, you also have contemplated adding more robots for handling various objects that require manipulation of more than two robots.

Acknowledgements. We wish to thank *the Centro de Investigación en Computación* of the IPN by the support to accomplish this project. R. Hernández thanks CONACYT by the scholarship received to complete his doctoral studies. Authors thank the European Union, the European Commission and CONACYT for the economical support. This paper has been prepared under economical support of the European Commission (grant FONCICYT 93829). The content of this paper is an exclusive responsibility of the authors and it does not reflect the position of the European Union. Finally, authors thank the reviewers for their comments for the improvement of this paper.

References

1. Antonelli, G., Arrichiello, F., Chiaverini S., Rao, K.J.: Preliminary Experiments of Formation Control using the Null-Space-Based Behavioral Control. Indian Institute of Technology, Guwahati, (2006)
2. Borenstein, J., Everett, H. R., Feng, L., Wehe, D.: Mobile Robot Positioning, Sensors and Techniques. Journal of Robotic Systems. 14, pp.231-249 (1997)
3. Cheng, L., Wang, Y.: A communication-based multirobot rigid formation control system: design and analysis. Int. J. Modelling, Identification and Control, Vol. 1, No. 1, (2006)
4. Renaud, P.: Vision-based mobile robot formation control. (2004)
5. Mead, R., Weinberg, J. B., Croxell, J. R.: A Demonstration of a Robot Control Algorithm and Platform. Illinois, USA (2007)
6. Lemay, M., Michaud, F., Létourneau, D., Valin, J.M.: Autonomous Initialization of Robot Formations. Quebec, Canada.

7. Li, X., Cai, Z.: Backstepping Based Multiple Mobile Robots Formation Control. In: IEEE International Conference on Intelligent Robots and Systems. Edmonton, Alberta, Canada (2005)
8. Castro, R., Alvarez J., Martinez, J.: Robot Formation Control using Backstepping and Sliding Mode Techniques. Department of Electrical Engineering, CINVESTAV-IPN, Mexico D.F., Mexico (2009)
9. Dixon, W.E., Dawson, D.M., Zergeroglu, E., Behal, A.: Nonlinear Control of Wheeled Mobile Robots. In: Thoma, M., Morari, M. (eds) Springer, USA (2001)
10. Sossa, J.H.: Feature Descriptors for Object Recognition. IPN, Mexico (2006)
11. Hu, M.K.: Visual Pattern Recognition by Moment Invariants. In: IRE Transactions on information Theory, pp. 179-187 (1962)
12. Casarrubias, H.: Trajectory Generation for Mobile Robots Khepera II Using Machine Learning Techniques. Thesis, CIC-IPN, Mexico (2007)
13. Cheng, L., Waing, Y.: A Communication-Based Multirobot Rigid Formation Control System: Design and Analysis. Modeling Identification and Control, Vol. 1. N. 1 (2006)
14. MATLAB 2010a, http://www.mathworks.com
15. JDK 6 with Netbeans, http://java.sun.com

Data Mining Algorithms

Cancer Detection using the *KDD* Process

Eddy Sánchez de la Cruz, Homero Alpuín Jiménez, and Pilar Pozos Parra

Universidad Juárez Autónoma de Tabasco,
División Académica de Informática y Sistemas,
Carretera Cunduacán-Jalpa Km. 1,
Cunduacán, Tabasco, Mexico
`eddsac@hotmail.com`, {`homero.alpuin,pilar.pozos`}`@dais.ujat.mx`

Abstract. Due to increasing deaths from breast cancer, it becomes necessary to incorporate computer technologies to support medical diagnosis. Despite this technological effort, the level of breast cancer detection is keeping low. This article presents the KDD process (*Knowledge Discovery in Databases*) as an alternative to obtain a trustable detection in medical imaging. Two important strategies that are used in this research are: *the Mejia's method*, which is used for data cleaning, and *LADTree algorithm*, which is used for detection. The implementation of the *KDD process* shows satisfactory results.

Key words: Breast Cancer, Mammograms, KDD Process, Detection.

1 Introduction

Data analysis tries to find trends or changes in data behavior. These trends and variations will be called, patterns. If the patterns are useful and relevant for the domain, then it is called knowledge.

At the beginning, the data analysis was performed manually using statistical techniques. Currently, this type of analysis is not feasible because of the large amount of information that can contain a modern database (DB) and for several data formats such as tables (Relational databases), sequences [1], graphs [12], images [8], and audio. In order to solve these problems the Data Mining arises (DM). DM is an automatic or semiautomatic process that seeks to discover hidden patterns in a dataset that are potentially useful for users of the DB [13], [1]. The main idea is to enhance and extract the information for interpretation and analysis by humans [10]. Extracting information from images is a complex process that uses different strategies to tackle the problem such as Artificial Neural Networks (ANN) [6], Bayesian networks [3], fuzzy logic [15], case based reasoning [3], bio-inspired algorithms [5] and hybrids of these.

In this article, the KDD process is used to analyze mammograms. For a specialist, the images are valuable to detect abnormalities; however, the specialist's analysis may be subject to errors, i.e. not guaranteed to find all the anomalies. Therefore, this research is use as a support in the doctor's decision to detect breast cancer. The rest of the article is divided as follows: Section 2 briefly explains the KDD process in this research. Section 3 discusses the first two phases of the KDD process. Section 4 explains the data transformation, data mining, and the pattern evaluation obtained, and then we test with a data set which shows a good detection of breast cancer. Finally, we conclude with a discussion for further work.

2 KDD Process

KDD is an interactive and iterative process which involves many steps and includes a lot of decisions that must be taken by the developer [9].

The KDD process for this research, where the steps are as follow:

- **Selection:** Given a set of different database takes the most representative.
- **Preparing data:** Data are filtered to reduce noise.
- **Data transformation:** The images are converted to table, taking the histogram of each one.
- **Data mining:** Applying an algorithm to obtain a pattern of behavior.
- **Assessment patterns:** Patterns are evaluated using a test database to determine whether breast cancer detection is appropriate.

3 Medical Images Preparation

Medical images analysis is a relevant issue for providing support to medical diagnosis. Some work has been done to detect abnormalities in mammographies using algorithms that implement different strategies [6], [3], [15], [3], [5] . Knowing that a mammography is an array of pixels, we want to find the pattern of the pixels that determines the existence of an anomaly, (see figure 1). The following sections will describe the first two phases of the KDD process: Selecting the database and preparing data.

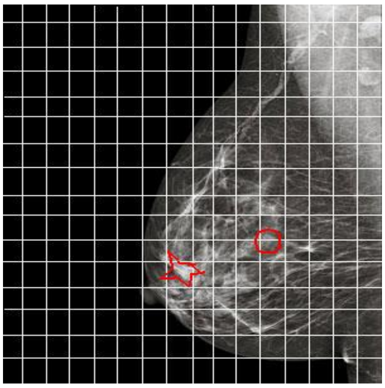


Fig. 1. Areas abnormalities.

3.1 Selecting the Database

We performed an investigation to obtain a public domain mammography's database and MIAS, which is a reduced database, was selected as it was used to test research in [7], [10] and which the South Florida University recommends in [4]:

Table 1. Database MIAS mammograms.

Name	Description
MIAS	Database created by the *Mammographic Image Analysis Society*, United Kingdom. Source: http://www.wiau.man.ac.uk/services/MIAS/MIASweb.html There is a reduced version that contains the same images as the original version, but reduced to a size of 1024 x 1024 pixels, available at http://peipa.essex.ac.uk/info/mias.html

3.2 Preparing Data

The image often contains information that is irrelevant for a given application. In such cases, the image must be filtered for removing all the irrelevant information.

All images have some amount of noise, which can be cause by the camera or the means of the signal transmission. Noise usually is manifest as isolated pixels that take a different gray level of its neighbors.

Filters are designed to reduce noise that can occur as a result of an image capture process, scanning or transmission [14]. This stage of KDD process uses the Mejía's filter proposed in [10]. This method is based on Non-subsampled Contourlet Transform (NSCT) and the Prewitt filter. The filter scheme is shown in figure 2.

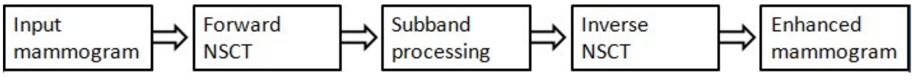


Fig. 2. Block diagram of the transform methods for images processing.

Describing in [10], this filtering process is efficiently used to enhancement a type of cancer, called micro-calcifications, in mammograms images. Figure 3 shows an image and the image after the filter. In the filtered image, we can see the enhancement of the brighter pixels which reveals the existence of micro-calcifications.

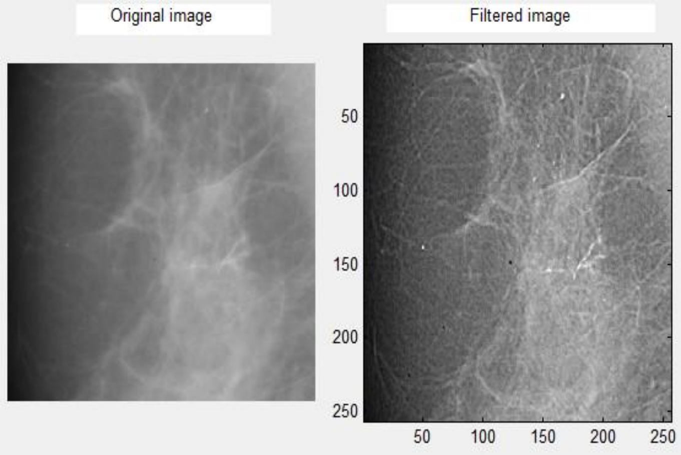


Fig. 3. NSCT Filter.

4 Processing and Analyzing Medical Images

In this section we explain the last three phases of the KDD process: Data transformation, Data mining and Patterns assessment.

4.1 Data Transformation

At this third phase of the KDD process, we obtained the histogram of each filtered mammography. The histograms describe frequency distribution of gray levels. The X axis represents the gray scale ranging from 0 to 255, and the Y axis represents the number of pixels that correspond to each gray level, an example of this type of histogram is shown in the figure 4.

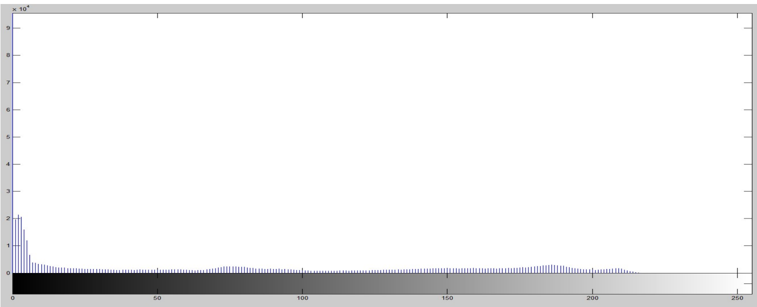


Fig. 4. Mammography's Histogram.

We take 100 instances (histograms) to build the training base, 50 with cancer and 50 without cancer. Of these 100 instances 13 are benign (B), 37 are normal (N) and 50 are malignant (M). Finally, in the data transformation, we build a file with the training base in ARFF format. An ARFF (= Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files have two sections. The first section is the *Header* information, which is followed the *Data* information [2]. The Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types [2], an example of this type of file is shown below:

```
% 1. Title: Mammograms Database EDDSAC
% 2. Sources:
% (a) Creator: Sanchez, Eddy
% (b) email: eddsacx@gmail.com
% (c) Date: June, 2010
%
@RELATION ima_training

@attribute Severidad {B,N,M} % B = Benign, N = Normal
 and M = Malignant
@attribute Pix0 numeric

.
.
.

@attribute Pix255 numeric
```

The Data of the ARFF file looks like the following:

```
@data
%---------------------------------------------------------------
% This histogram corresponds to an image
% which presents a benign abnormality (B).
%---------------------------------------------------------------
B,520417,26413,16747,14412,11561,9346,10517,
7801,5836,5303,4872,4381,4115,3718,3575,
3153,2767,2609,2424,2350,2235,2126,2226, ...
```

4.2 Data Mining

To implement the pattern detection we use the software WEKA (*Waikato Environment for Knowledge Analysis*) v. 3.6.1. It is a collection of machine learning algorithms for data mining tasks. It contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes [11]. Weka is open source software issued under the GNU General Public License [11].

Weka contains different classification strategies, such as: bayesian algorithms (*bayes*), classification functions (*functions*), lazies algorithms (*lazy*), committee algorithms *meta*), multi-instance learn algorithms (*mi*), rules generation algorithms (*rules*), trees generation decision algorithms (*trees*), and miscellany algorithms *misc*). We tried with the classifiers of each strategy and the best result was obtained with the *Decorate classifier*. Decorate classifier is a meta-learner, classified within committee strategies. Decorate classifier can use different base classifiers. In this research we use as base classifier the multiclass decision tree: LADTree. This base classifier uses the algorithm LogitBoost, the algorithm is as follows:

Algorithm 1.1: Algorithm LogitBoost.

```
Result: If $p(1|a) > 0.5$ predict the first class, if not the second
begin
    for $j = 1$ until $t$ do
        for $a[i]$ do
            Assign the objective value for the regression to
            $z[i] = (y[i] - p(1|a[i]))/(p(1|a[i]) * (1 - p(1|a[i]))$
            Assign the weight of the instance to
            $[i] = p(1|a[i]) * (1 - p(1|a[i])$
        Fit a regression model $f_j$ at datas with class values $z[i]$ and weight $w[i]$

end
```

4.3 Patterns Assessment

For testing, we used four dataset of 20 instances each one. The following tables show the results:

inst#	actual	predicted	error	inst#	actual	predicted	error
1	1:B	1:B		11	2:N	2:N	
2	1:B	1:B		12	3:M	3:M	
3	2:N	2:N		13	2:N	2:N	
4	2:N	2:N		14	1:B	1:B	
5	3:M	3:M		15	2:N	2:N	
6	2:N	2:N		16	3:M	3:M	
7	2:N	2:N		17	3:M	3:M	
8	1:B	1:B		18	2:N	2:N	
9	2:N	2:N		19	1:B	1:B	
10	3:M	3:M		20	2:N	1:B	+
Correctly Classified 19 < −− 95%							
Incorrectly Classified 1 < −− 5%							

In the first dataset, of 20 instances 5 are benign (B), 10 normal (N) and 5 malignant (M). We obtained 95 % correctly classified instances, i.e. 19 of 20.

inst#	actual	predicted	error		inst#	actual	predicted	error
1	2:N	2:N			11	3:M	3:M	
2	3:M	3:M			12	2:N	2:N	
3	3:M	3:M			13	2:N	2:N	
4	2:N	2:N			14	1:B	1:B	
5	1:B	1:B			15	2:N	2:N	
6	2:N	1:B	+		16	3:M	3:M	
7	1:B	1:B			17	2:N	2:N	
8	1:B	1:B			18	3:M	3:M	
9	2:N	2:N			19	1:B	1:B	
10	2:N	2:N			20	1:B	1:B	

Correctly Classified 19 < −− 95%
Incorrectly Classified 1 < −− 5%

In the second dataset, of 20 instances 6 are benign (B), 9 normal (N) and 5 malignant (M). We obtained 95 % correctly classified instances, i.e. 19 of 20.

inst#	actual	predicted	error		inst#	actual	predicted	error
1	1:B	2:N	+		11	3:M	3:M	
2	2:N	2:N			12	3:M	3:M	
3	3:M	3:M			13	2:N	2:N	
4	2:N	2:N			14	1:B	1:B	
5	2:N	2:N			15	1:B	1:B	
6	2:N	2:N			16	3:M	3:M	
7	3:M	3:M			17	2:N	2:N	
8	2:N	2:N			18	3:M	3:M	
9	1:B	3:M	+		19	1:B	1:B	
10	2:N	2:N			20	1:B	1:B	

Correctly Classified 18 < −− 90%
Incorrectly Classified 2 < −− 10%

In the third dataset, of 20 instances 6 are benign (B), 8 normal (N) and 6 malignant (M). We obtained 90 % correctly classified instances, i.e. 18 of 20.

inst#	actual	predicted	error		inst#	actual	predicted	error
1	1:B	1:B			11	2:N	2:N	
2	2:N	2:N			12	2:N	2:N	
3	3:M	3:M			13	3:M	3:M	
4	3:M	3:M			14	2:N	2:N	
5	3:M	3:M			15	2:N	2:N	
6	2:N	2:N			16	2:N	2:N	
7	2:N	2:N			17	2:N	2:N	
8	3:M	3:M			18	2:N	2:N	
9	2:N	2:N			19	2:N	2:N	
10	2:N	2:N			20	1:B	1:B	

Correctly Classified 20 < −− 100%
Incorrectly Classified 0 < −− 0%

Finally, in the fourth dataset, of 20 instances 2 are benign (B), 13 normal (N) and 5 malignant (M). We obtained 100 % correctly classified instances, i.e. 20 of 20.

We have four incorrectly classified instances (see table 2). 2 instances are benign (B) and 2 are normal (N). Instances malignant (M) indicate that breast cancer does exist. In table 2 we can see that none malignant instance is incorrectly classified, which implicates that the classificator performs a correct diagnoses of all malignant instances as shown in table 3.

Given above results we are motivated to use this classificator with another mastography databases in order to verify the correct performance of our classificator.

Table 2. Instances incorrectly classified.

dataset	inst#	actual	predicted	error
1	20	2:N	1:B	+
2	6	2:N	1:B	+
3	1	1:B	2:N	+
3	9	1:B	3:M	+

Table 3. Classification of M instances.

dataset	inst#	actual	predicted	err	dataset	inst#	actual	predicted	err
1	5	3:M	3:M		3	7	3:M	3:M	
1	10	3:M	3:M		3	11	3:M	3:M	
1	12	3:M	3:M		3	12	3:M	3:M	
1	16	3:M	3:M		3	16	3:M	3:M	
1	17	3:M	3:M		3	18	3:M	3:M	
2	2	3:M	3:M		4	3	3:M	3:M	
2	3	3:M	3:M		4	4	3:M	3:M	
2	11	3:M	3:M		4	5	3:M	3:M	
2	16	3:M	3:M		4	8	3:M	3:M	
2	18	3:M	3:M		4	13	3:M	3:M	
3	3	3:M	3:M						

5 Conclusions and Future Work

In this research we combined the *Mejía's filtering method* and the *Decorate classificator* in the *KDD process* to detect cancer in digital mammograms; the results show a satisfactory detection of breast cancer using the MIAS database.

However, as future work we want to get another mammogram databases to repeat the test and corroborate that this approach provides a good breast cancer detection in general cases.

References

1. Sanghamitra Bandyopadhyay, Ujjwal Maulik, Lawrence B. Holder, and Diane J. Cook. Advanced methods for knowledge discovery from complex data. 2005. Springer.
2. Remco R. Bouckaert, Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann, Alex Seewald, and David Scuse. Weka manual for version 3-6-0. 2008. University of Waikato, Hamilton, New Zealand.
3. Ernesto Coto. Methods for medical image segmentation. 2003. Springer-Verlag.
4. DDSM. Ddsm: Digital database for screening mammography. 2009. University of South Florida Digital Mammography Home Page http://marathon.csee.usf.edu/Mammography/Database.html.
5. Eddy Sánchez de la Cruz and Jorge P. Torres. Sia in the diagnosis of cancer. 2009. National Congress of Computers and Systems (CONAIS). Villahermosa, Tabasco, México.
6. N. D. Duque, J. C. Chavarro, and R. Moreno. Seguridad inteligente. 2007. Science Et Technica , Vol. 13, No. 35.
7. Ahmed Farag and Samia Mashali. Dct based features for the detection of microcalcifications in digital mammograms. 2004. Univ of Texas at El Paso. IEEE.
8. Morales Gonzalez and Aurora B. Image feature extraction of bone marrow cells for the classification of acute leukemias. Master's thesis, 2006. National Institute of Astrophysics, Optics and Electronics.
9. José Molina López and Jesús García Herrero. *TECHNICAL ANALYSIS OF DATA - PRACTICAL APPLICATIONS USING MICROSOFT EXCEL And WEKA*. Universidad Carlos III from Madrid, 2004.
10. José M. Mejía Mu noz. The nonsubsampled contourlet transform for enhancement of microcalcifications in digital mammograms. 2009. 8th Mexican International Conference on Artificial Intelligence MICAI-2009. Guanajuato, México.
11. WEKA University of Waikato. Weka 3: Data mining software in java. 2010. http://www.cs.waikato.ac.nz/ ml/weka/.
12. Iván Olmos, Jesús González, and Mauricio Osorio. Mining common patterns on graphs. 2005. International Conference on Computational Intelligence and Security, Lecture Notes in Artificial Intelligence, Vol. 3802, 41-48, Springer Verlag.
13. Abraham Silberschatz, Henry Korth, and S. Sudarshan. *Database Fundamentals.* McGrawHill, 4a Ed., 2002.
14. Universidad-Jaén. Noise reduction in a digital image. 2006. Electronic Engineering Department. Area Systems and Automation Engineering. Pg. 2-7.
15. César Cardona Valencia. Evaluation of algorithms based on fuzzy logic applied to the preprocessing and edge detection in digital images, 2004.

Fuzzy Logic Algorithms

Extending Jess with Type-2 Fuzzy Logic

Carelia Gaxiola-Pacheco, Dora-Luz Flores, Manuel Castañón-Puga, Antonio Rodríguez-Díaz, Juan-Ramón Castro, and Iván Espinoza-Hernández

Universidad Autónoma de Baja California,
Tijuana, Baja California, Mexico
{cgaxiola,dflores,puga,ardiaz,jrcastror,espinoza.ivan}@uabc.edu.mx

Abstract. Jess (Java expert system shell) has been used as an inference system to model decision-making in multi-agent systems (MAS). In addition, fuzzy inference systems (FIS) have been mostly used to regulate, control and implement decision-making systems. To extend its functionality, we present an extension to Jess applying interval type-2 FIS. We conclude with a case of study as an example.

Keywords: Java, Expert System, Type-2 Fuzzy Logic.

1 Introduction

Traditional artificial intelligence approaches are limited in developing autonomous robots and social simulations, however fuzzy logic, neural networks, genetic algorithms, and symbolic processing may be useful to improve their performance. While machine learning is considered probably the most difficult task to improve, collaboration between robots or agents is also a crucial feature to consider. These types of systems will be most useful when many of these features are networked together, but if getting one system to intelligently operate in the real world is extremely challenging, then getting several of them to work together is even more difficult [1]. From a computational point of view, fuzziness and distribution can be tackled with hybrid intelligent software agents. The use of fuzzy logic in agents has previously shown satisfactory results [2]. Computational models such as AvatarSim, FLAME, and PETEEI use type-1 fuzzy logic to simulate emotions and personality in agents [3].

1.1 Type-2 Fuzzy Logic

A type-2 fuzzy logic (T2FL) tool was developed as a toolbox for Matlab by [4] and has been used for control applications [5], edge detection in digital images [6], and more. Castro et al. show the creation of a fuzzy inference system using this tool in [7]. A novel tool for simulations based on agents named Wiinik has been developed [8], where an agent's behavior is determined by an interval type-2 fuzzy inference system (IT2FIS), which is designed to define the agent's psychological elements, semantic networks are used to represent the information acquired by the agent from its environment, allowing the agent to communicate in a more meaningful way. Semantic networks, although not standardized, have great potential to model any kind of knowledge, regardless of an agent's capability.

1.2 Fuzzy Jess

The Integrated Reasoning Group of the Institute for Information Technology of the National Research Council of Canada has developed two fuzzy logic software packages: FuzzyCLIPS and FuzzyJ [9]. FuzzyCLIPS is an extension of the CLIPS (C Language Integrated Production System), which is an expert system shell developed by NASA. It integrates a fuzzy reasoning engine to CLIPS facts [10]. The FuzzyJ Toolkit is a Java API (Application Program Interface) for handling fuzzy logic systems. It can be used as a standalone system or integrated with Jess (FuzzyJess). FuzzyJess provides similar capabilities but it is more flexible than FuzzyCLIPS. Inputs and outputs can be handled using Java APIs. FuzzyJess is the rule engine used for implementing fuzzy sets and the reasoning mechanism used to simulate an agent's performance. Epstein et al. and Martínez et al. present tools which allow simulations in a MAS where agents decide according to fuzzy-logic rules in [11, 12] .

1.3 Multi-Agents Systems

In recent years, the development of computational capabilities has redefined the way in which philosophers, social scientists and other researchers see the world. Scientific enquiries have generally shifted from paradigms based on a linear conception of the world to a nascent, more holistic view based on non-linearity. The distinction between linear and non-linear science has not been fully resolved in the literature [13], but as a working definition for this paper, we can say that the linear paradigm of the past is based on the assumption of independence of whatever atoms or agents are used to describe reality through a given model. In stark contrast, the growing non-linear view of the world is based on a perception of ingrained interdependence. The most common way of describing the difference between the linear and non-linear paradigms is that in the former the aggregate is equal to the sum of its parts, while in the latter the aggregate is more than the sum of its parts [14].

Simulation has been adopting multi-agent systems as one of the most promising ways to achieve social computational models. Agent-based models facilitate the analysis and design of models, once that agents' behavior is part of a theoretical system, this behavior can be computationally implemented. Then the language of agents from the point of view of analysis and the language of agents from the point of view of design are very similar, but not necessarily equal.

From the scientific point of view, MAS depend on the available methodology: the easiness offered by the analysis of actor and context to construct theoretically agents. MAS from the computational point of view, depend on the available technology, this means, the easiness offered by the software and hardware design to generate software agents [15].

2 Extending Jess with Type-2 Fuzzy Logic

The addition of fuzzy logic to Jess is not new, existing extensions such as FuzzyJess provide a traditional type-1 fuzzy extension, allowing the rules to be written as fuzzy rules and providing a variety of membership functions and modifiers to describe linguistic variables. While linguistic variables provide a simple and appropriate way to represent a range of values that have different degrees of belonging, these ranges must be given a single, precisely defined value. T2FL extends fuzzy sets by allowing a range of values to define the range of belonging of a given element in a fuzzy set. By modeling the uncertainty on a fuzzy set we are able to widen our definitions or more precisely model our views on what does and does not belong in a concept.

When modeling real world abstract concepts such as hot, cold, fast or slow, using a type-1 membership function can be made to fit the concept but it will be tied to the views of a very small set of opinions, what is considered hot for some might be considered not so hot or even cold for others. With type-2 fuzzy sets we are able to represent this disparity in opinion for each concept. Depending on the application, this advantage can be used to give greater margin of acceptance when processing signals with noise, recognising imagery with a neuro-fuzzy network, representing the opinions of a group on certain issues or defining abstract concepts in natural language processing.

In this paper we discuss the possibility of using T2FL in conjunction with a rule-based system to control the behavior of computer agents. Agent models adhering to the production rule system will need a rule interpreter. Among the many options Jess presents itself as simple and adequate, its features are sufficient for most tasks and when it would seem that we reach a limit to what we can do, it is also capable of handling extensions.

Using existing methods, T2FL comes with a considerable performance cost. As it is a relatively new approach, we hope that this issue will be resolved by the introduction of efficient algorithms and the increase in computational capabilities. Taking into account this limitation, our work focuses on the potential modeling benefits rather than on immediate practical usage.

2.1 Adding Interval Type-2 Fuzzy Inference System to Jess

Using type-2 fuzzy extension with Jess is as simple as importing a library and using Java objects. The extension is self contained in a Java JAR file, few user functions are defined to facilitate creating fuzzy sets and it is mainly handled as a typical Java object. What this means is that, unlike FuzzyJess, using this extension does not require re-engineering any rules to conform to a fuzzy rule and it also does not require the addition of any code to the standard Jess interpreter.

Inputs and outputs are added to this object which will be used by a set of rules that must also be defined. The general layout in creating a working FIS is in essence exactly the same as any common type-1 inference system, the only notable difference will be defining membership functions and the defuzzified value obtained for outputs. Instead of returning a single numerical value, we can obtain three values: a leftmost point, a rightmost point, and a centre point value.

The following code demonstrates creating a FIS object, adding one input and one output, and two linguistic variables for each:

```
(bind ?fis ((new Fis))
(call ?fis addInput "Time")
(call ?fis addToInput "Time" "Fast" (IGaussMT2 ?ls1 3.0 0.6 1.4))
(call ?fis addToInput "Time" "Slow" (IGaussMT2 ?ls1 3.0 9.6 10.4))
(call ?fis addOutput "Score")
(call ?fis addToOutput "Score" "0" (ITriT2 ?ls4 -1 0 1 -1.2 0.2 1.2))
(call ?fis addToOutput "Score" "1" (ITriT2 ?ls4 0 1 2 0.2 1.2 2.2))
```

2.2 Adding Fuzzy Rules into Extension

Once a FIS object has been created and at least one input and one output have been added, it will now be possible to add the rules that will determine its behavior. Adding a rule is demonstrated with the following code:

```
(?fis addRule "[Time:Slow] [Cost:Cheap] [Capacity:Low]" "[Score:1]")
```

To add a rule, the method requires two formatted strings as inputs. The first string will identify linguistic variables for inputs, each set of input and linguistic variable surrounded by brackets and each input name and linguistic variable separated by a colon. The seconds string follows the same pattern to identify outputs and their linguistic variables.

3 Case of Study

We illustrate our proposal with a simple case taken from [16, 17] where a planning mechanism is required to schedule tasks within a community of autonomous agents. Each agent represents a production unit in a factory and some agents must negotiate task allocation according to available resources. Figure 1 shows the structure of a given organisation, which involves at least three agent types: Production Planning Agent (PPA), Production Managing Agent (PMA) and Production Agent (PA).

Several agents can be created with PPA, PMA and PA roles. Each agent can use a FIS to resolve input values depending on the assigned role. Agent interactions create a dialogue between roles where each agent constantly make decisions depending of the plan and the tasks to be processed.

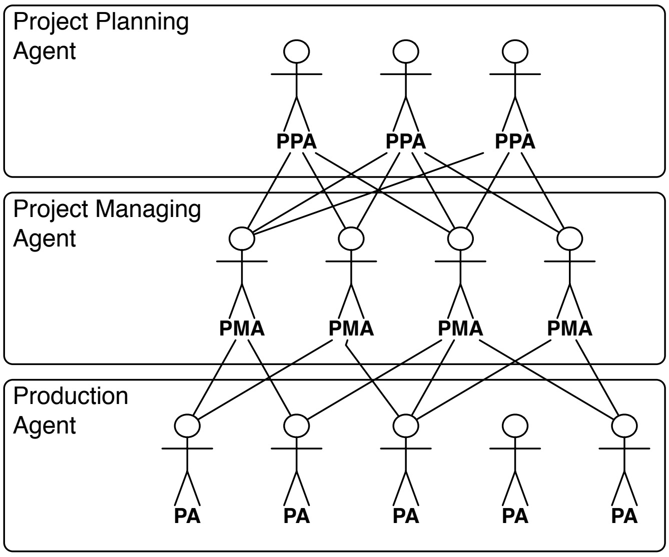


Fig. 1. Each PA represents a production unit in a factory. A PMA must negotiate the delegation of work and decomposition of tasks according to available resources in order to meet production schedules set by the PPA.

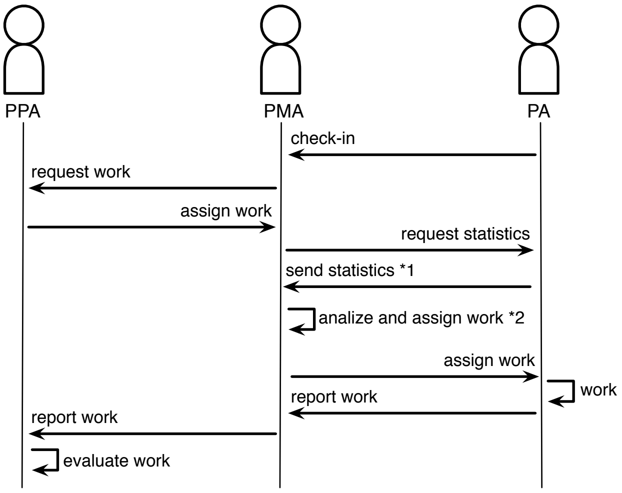


Fig. 2. Script flow Company.

3.1 Fuzzy Decision Making System into Organizational Structure

In this scheduling example, the PPA is in charge of project planning. The PMA performs project management in terms of contracting the best possible PA (in terms of operational costs, the delivery time and current capacity availability). The PA represents the lowest level production unit that simulates or encapsulates shop floor production process.

We will compare the case where the PMA must decide which PA seek to assign new tasks according to their production capacity.

Nonfuzzy approach with Jess In the nonfuzzy logic approach, the PMA can opt for an election where the PA has an absolute value at the time of production in comparative with others.

For example, the following rule represents the decision of PMA to choose the PA within available resources.

```
(bind ?score1 ( + (/ ?co1 20) (/ (- 10 ?ti1) 10) (/ ?ca1 3)))
(bind ?score2 ( + (/ ?co2 20) (/ (- 10 ?ti2) 10) (/ ?ca2 3)))
(if (< ?score1 ?score2) then (return 1))
(if (> ?score1 ?score2) then (return -1))
(return 0)
```

Each variable is given equal weight by normalising its value and adding them giving a maximum value of three for the best option and a minimum of zero for the worst option. This process allows each agent to be given a score and based on this value, they can be arranged in descending order making the best option the first in the list.

Type-1 Fuzzy Approach with FuzzyJess In a fuzzy approach, the PMA will not receive a numerical value but a fuzzy set as input. From this it must determine the score in a similar fashion as the previews case using fuzzy rules. The inputs, outputs and rules are set up as it is described in Table 1.

Table 1. PMA linguistic variables and values.

Rule	Time	Cost	Capacity	Score
r1	slow	cheap	low	1
r2	slow	cheap	high	2
r3	slow	expensive	low	0
r4	slow	expensive	high	1
r5	fast	cheap	low	2
r6	fast	cheap	high	3
r7	fast	expensive	low	1
r8	fast	expensive	high	2

The following code demonstrates how a type-1 fuzzy rule would look like using FuzzyJess. It should be noted that they are written in the similar form as Jess rules, making it necessary to mix fuzzy rules with nonfuzzy rules for our case of study where we control an agents behavior with rules.

```
(defrule r1
(theTime ?ti& :(fuzzy-match ?ti "slow"))
(theCost ?co& :(fuzzy-match ?co "cheap"))
(theCapacity ?ca& :(fuzzy-match ?ca "low"))
->
(assert (theScore (new nrc.fuzzy.FuzzyValue ?*scoreFvar* "1")))
)
```

Example of the same rule with the proposed extension would be written as:

```
((?fis addRule "[Time:Slow] [Cost:Cheap] [Capacity:Low]" "[Score:1]")
```

Type-2 fuzzy Approach with T2FIS Extension In a type-2 fuzzy approach, the PPA must not only choose an agent without an accurate production time, but also where each linguistic variable involves a degree of uncertainty, and will therefore have to interpret not only the evaluation but consider the degree of uncertainty in each assessment.

For example, the following rules represent the decision of a PMA to choose the best PA within the available options.

```
(bind ?score1 (?dfscore1 getXcoa))
;consider range of uncertianty in answer
(bind ?dif1 (- (?dfscore1 getXl) (?dfscore1 getXr)))
(bind ?score2 (?dfscore2 getXcoa))
;consider range of uncertianty in answer
(bind ?dif2 (- (?dfscore2 getXl) (?dfscore2 getXr)))
(if (> ?dif1 ?dif2) then
(bind ?score2 (+ ?score2 0.3)))
(if (< ?dif1 ?dif2) then
(bind ?score1 (+ ?score1 0.3)))
```

Uncertainty is not given as much weight as the three variables since a small amount of uncertainty is preferred to a long amount of time for job completion for this example. There are many factors to consider when determining how to weight each variable, every job will have different needs and there must be a certain tolerance level for uncertainty when dealing with real scheduling problems.

Figure 3 shows PA's outputs using different approaches: nonfuzzy, type-1 and type-2 fuzzy logic. Uncertainty can be observed in T2FL output. Compared with nonfuzzy output, which shows only a single value as a result, a whole range of values can be obtained using a fuzzy approach, leading to a a more realistic estimate of an area where our actual value may be.

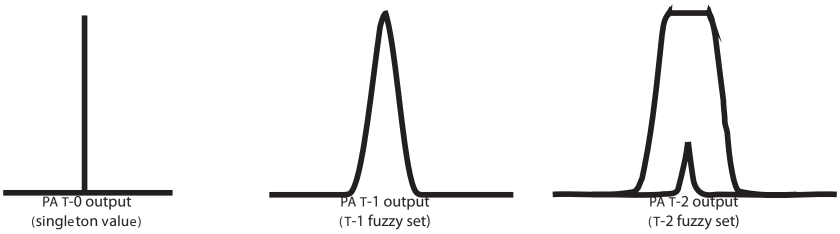


Fig. 3. PA's outputs: nonfuzzy, type-1 and type-2 fuzzy logic.

4 Conclusions and Future Work

It is possible to extend the functionality of Jess to introduce a type-2 fuzzy inference system. We used the example of Planning-Management-Resource to validate its capability.

Current work is focused on type-2 fuzzy inference systems for modeling real cases in order to explore its application.

As future work the possibility of extending Jess with a learning system to strengthen the system of type-2 fuzzy inference is being considered, so that agents change their behavior by evaluating, modifying and/or adding new rules in its inference system to build agents that adapt to the circumstances.

The vision for the future is to have a software agent model with an intelligent hybrid approach allowing better representation of MAS.

References

1. Long, L.N., Hanford, S.D., Janrathitikarn, O., Sinsley, G.L., and Miller, J.A.: A Review of Intelligent Systems Software for Autonomous Vehicles. In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications, (2007)
2. Sharma, S., Singh, H., and Prakash, A.: Multi-agent modeling and simulation of human behavior in aircraft evacuations. IEEE Transactions on Intelligent Transportation Systems, 44(4), (2008) 1477–1488
3. El–Nasr Seif, M., and Yen, J.: Agents, Emotional Intelligence and Fuzzy Logic. In: Proceedings of NAFIPS. Florida, US (1998)
4. Castro, J.R., Castillo, O., and Martínez, L.G.: Interval Type-2 Fuzzy Logic Toolbox, Engineering Letters, 15(1), (2007) 89–98
5. Castro, J.R., Castillo, O., and Melin, P.: An Interval Type-2 Fuzzy Logic Toolbox for Control Applications, FUZZ-IEEE, (2007), 1–6

6. Mendoza, O., Melin, P., and Castro, J.R.: The Use of Interval Type-2 Fuzzy Logic as a General Method for Edge Detection. In: IFSA/EUSFLAT Conf., (2009) 774–779
7. Castro, J.R., Castillo, O., Melin, P., and Rodríguez–Díaz, A.: Building Fuzzy Inference Systems with a New Interval Type-2 Fuzzy Logic Toolbox. Transactions on Computational Science, vol 1, (2008) 104–114
8. Flores, D.L., Rodríguez–Díaz, A., Castro, J.R., and Gaxiola, C.: TA-Fuzzy Semantic Networks for Interaction Representation in Social Simulation. Evolutionary Design of Intelligent Systems, SCI, Springer Berlin / Heidelberg (2009) 257–270
9. Firedman-Hill, E.J.: Java Expert System Shell. Available from http://herzberg.ca.sandia.gov/jess. (1998)
10. $http://www.iit.nrc.ca/IR_public/fuzzy/$
11. Epstein J.G., Michael Mohring,M., and Troitzsch, K.G.: Fuzzy-Logical Rules in a Multi-Agent System. In: SimSocVI Workshop, Groningen, (2003) 25
12. Martínez–Miranda, J., Aldea, A., and Bañares–Alcántara, R.: Agent Based Simulation in the Selection of Work Teams. Computación y Sistemas. 7(3) ISSN 1405-5546. (2004) 210–223
13. Abbott, R.: Putting complex systems to work. Complexity, 13(1) (2007) 30–49
14. Bar–Yam, Y.: Dynamics of Complex Systems, Addison–Wesley, Reading, MA, (1997)
15. Suarez, E.D., Rodríguez-Díaz, A., and Castañón-Puga, M.: Fuzzy agents. In: Castillo O, Melin P, Kacprzyk J, Pedrycz W (eds) Soft computing for hybrid intelligent systems, vol 154. Studies in computational intelligence. Springer, Berlin / Heidelberg (2008)
16. Pechoucek, M., Riha, A., Vokrinek, J., Marik, V., and Prazma, V.: Explantech: Applying multi-agent systems in production planning. International Journal of Production Research 40 (15):(2002) 3681–3692
17. Pechoucek, M., Rehak, M., Charvat, P., Vlcek, T., and Kolar, M.: Agent-based approach to mass-oriented production planning: Case study. In: IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 37 (3):(2007) 386–395

Feature Extraction in Medical Image Applications using Fuzzy Logic

Fabián Torres-Robles, Francisco J. Gallegos-Funes,
Alberto J. Rosales-Silva, and Rene Cruz-Santiago

Mechanical and Electrical Engineering Higher School,
National Polytechnic Institute,
Av. IPN s/n, Edificio Z, 3$^{\underline{er}}$ piso; SEPI-Electronica,
Col. Lindavista, 07738, México D. F., Mexico
dae.wong@gmail.com, fgallegosf@ipn.mx, arosaless@ipn.mx

Abstract. In this paper, we present the results to apply a fuzzy feature extraction method for medical imaging. The application is in the detection of cardiac insufficiency by using thorax image radiographies. From simulation results we observe that the proposed fuzzy method improves the feature extraction in comparison with a classical method.

Keywords: Fuzzy Feature Extraction, Cardiac Insufficiency.

1 Introduction

Digital image processing has been a useful tool to help doctors in diagnosis of several diseases. Digital chest radiographies have been studied and analyzed by image processing systems since two decades ago. The most common application of these image processing systems is the diagnosis of lung tumors. Chest radiographies are also used by doctors to diagnose several heart conditions, such as cardiac insufficiency.

The number of people suffering of cardiac insufficiency increases every year. The number of cases in Mexico increases 10% per year. In a paper in the "Revista Mexicana de Cardiología" shows that in Mexico there are about 750 thousand patients affected by cardiac insufficiency, this number increases about 10% per year, which means that there are 75 thousand new cases each year [1].

New alternative methods of diagnosis are needed to be developed in countries like México, so the non experts in cardiology can make a reliable diagnosis and start a preventive treatment to patients who suffer of cardiac insufficiency, until they are able to treat the condition with a cardiologist [2, 3].

The objective of this paper is to develop an algorithm to extract several features that can make a reliable diagnosis of cardiac insufficiency by analyzing thorax digital radiographies. The experimental results show that the proposed fuzzy method improves the feature extraction in comparison with a classical method.

© G. Sidorov et al. (Eds.)
Advances in Soft Computing Algorithms
Research in Computing Science 49, 2010, pp. 131-141

2 Detection of Cardiac Insufficiency

A cardiac insufficiency is defined as a clinic syndrome in which the anomalies in the heart structure and function cause the malfunction and incapacity of this organ to expel blood or refill with blood at the rate needed by other organs to function. There are several classifications of cardiac insufficiencies, here we present the must common classifications [2, 3]:

a) Right insufficiency. This insufficiency is allocated on the right side of the heart. This ventricle loses the function of pumping blood and it can be retained by other organs causing congestion.

b) Left insufficiency. It is allocated on the left side of the heart; the left side receives blood from the lungs and is in charge of pumping blood to the rest of the body. If the left side of the heart fails the rest of the body can not receive sufficient oxygen causing fatigue, and blood can be retained by the lungs.

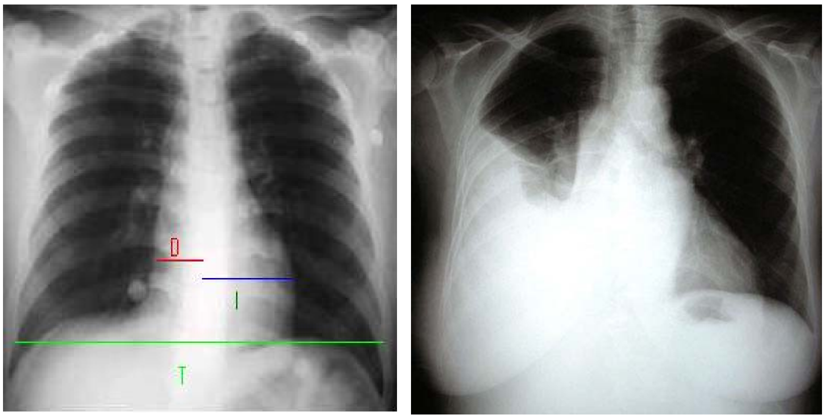
The first stage to give a reliable diagnose by analyzing thorax radiography is to evaluate if the radiography is well taken. A bad radiography can give us false information about the parameters that can be found to detect and diagnose the insufficiency. Listed below are some of the parameters and anomalies that are shown in a thorax radiography that can help us to diagnose a cardiac insufficiency.

Cardiomegaly refers to the abnormal growth of the heart. This condition is caused by the excessive work of the heart that it has to perform to function properly, just like a muscle; the heart increases its size and strength when it is forced continuously. Cardiomegaly can be identified by measuring of the cardiothoracic index (IC) [2],

$$IC = (D+I) / T \quad \textbf{(1)}$$

where “D” is the maximum distance between the mid line of the thorax and the right border of the heart, “I” is the maximum distance between the mid line of the thorax and the left border of the heart, and “T” is the maximum distance between the left and right borders of the thorax. According to Fig. 1a, if “IC” is greater than 0.5, this indicates cardiomegaly in most cases.

Pleural edema is an accumulation of fluid between the membrane covering the lungs and the thoracic cavity (see Figure 1b). This can be caused primarily by left cardiac insufficiencies; when the left side of the heart fails the blood is retained in the lungs, when there is too much blood retained this causes edema. When the edema is not severe can be seen as a decrease in the angle between the lung and the diaphragm.



a) b)

Fig. 1. Parameters and anomalies in a thorax radiography to diagnose a cardiac insufficiency. a) Measurement of Cardiothoracic index, b) Right pleural edema.

3 Proposed Method

The proposed method has three basic stages: pre-processing, segmentation and feature extraction [4, 5]:

a) Pre-processing: This process modifies the intensity of the pixels of an image to change the contrast between them; this is done to obtain a better image with the intensities needed in the proposed algorithm. We mention the methods to achieve this**.**

1. Adjustment. This method consists in the use of a threshold between intervals of intensities. The intensities between the limits are mapped into new intensity levels define in a curve that has been chosen before [6].
2. Histogram equalization. A histogram is a vector that contains the number of pixels that has the same level of intensity. Histogram equalization consists in increasing the dynamic range of an image and gives it a better contrast. The algorithm is given in the next steps [6],
    a. Calculate the image histogram.
    b. Calculate the cumulative distribution function vector (*cdf*), adding the value of *cdf(i)* with all the values before *i* where *i*=1,2,3,4,….*n*, and *n* is the number of grey levels in image.
    c. The general histogram equalization formula is:

$$h(v) = round\left\{ \frac{cdf(v) - cdf_{\min}}{(M \times N) \times cdf_{\min}} \times (L-1) \right\} \quad \textbf{(2)}$$

where cdf_{min} is the minimum value of the cumulative distribution function, $M \times N$ gives the number image pixels, and L is the number of grey levels used in the image. This equation is used to obtain a new histogram with a wider dynamic range**.**

b) Segmentation: Segmentation refers to the process of partitioning a digital image into multiple segments. Segmentation algorithms used in this paper involve the edge detection and mathematical morphology [4]:

1. Edge detection. The edge detection is based on finding the areas in an image that have the maximum contrast. First level operators are approximations of the first order derivate of the image. The first order derivate shows a maximum where the intensity levels change quickly; this is the point where the maximum contrast is found.
   Canny algorithm. This is the most efficient first order algorithm of edge detection. Canny algorithm steps are as follows,
   - Gaussian Filter
   - Sobel operator. It consists of two matrices which are approximations of the first order derivative in vertical and horizontal directions,

$$Mx = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad My = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \textbf{(3)}$$

   - Non-max suppression. Eliminate the non maximum pixels in a window.

- Hysteresis threshold.

2. Mathematical morphology: Mathematical morphology processes images into shapes. To carry out transformations are structural elements consisting on matrices of 0´s and 1´s. The structural element can have any form needed by the program. There are two basic transformations: the dilation operator, (which grows or thickens some objects in a binary image), and the erosion operator (which shrinks or thins some objects in a binary image) [7].

c) Feature extraction: Feature extraction consists in extracting the characteristics needed by the program. Sometimes the characteristics are not easy to obtain from a segmented image, so it is useful to implement some fuzzy logic processing to achieve a reliable measurement of the feature [5].

1. Fuzzy logic: Fuzzy systems are useful to solve problems in which some data can have a level of uncertainty and imprecision [8,9]. These systems take decisions based on levels of membership between [0, 1] to a group, not like binary logic systems which only take levels of 1 or 0, meaning it belongs or not belongs. These systems deal with problems where the borders are not well defined in which a variable is part of a certain group, so this variable has a membership function that defines the level of membership of the variable to a certain group. The membership functions that represent the membership value of a pixel in the radiography to a part of the thorax are defined by eqs. (4), (5), and (6). Fig. 2 shows the membership functions. With these membership functions is feasible to create fuzzy rules that help us determine if the maximum distance (D and I defined before) found really belongs to the heart or belongs to another part of the body (like the diaphragm or the trachea). If the maximum measurement is not part of the heart, the program should take another measurement until it obtains the maximum measurement that is part of this.

$$trachea(p) = \begin{cases} 1, & if \quad p \le \bar{p}_T - \sigma_T \\ 1 - (p + \bar{p}_T - \sigma_T / 2\sigma_T), & if \quad \bar{p}_T - \sigma_T < p < \bar{p}_T + \sigma_T \\ 0, & otherwise \end{cases} \tag{4}$$

$$heart(p) = \begin{cases} 1, & if \quad \bar{p}_T + \sigma_T \le p \le \bar{p}_D - \sigma_D \\ 1 - (p + \bar{p}_D - \sigma_D / 2\sigma_D), & if \quad \bar{p}_D - \sigma_D < p < \bar{p}_D + \sigma_D \\ (p - \bar{p}_T + \sigma_T / 2\sigma_T), & if \quad \bar{p}_T - \sigma_T < p < \bar{p}_T + \sigma_T \\ 0, & otherwise \end{cases} \tag{5}$$

$$diaphragm(p) = \begin{cases} 1, & if \quad p \ge \bar{p}_D + \sigma_D \\ (p - \bar{p}_D + \sigma_D / 2\sigma_D), & if \quad \bar{p}_D - \sigma_D < p < \bar{p}_D + \sigma_D \\ 0, & otherwise \end{cases} \tag{6}$$

where σ_D is the standard deviation of the diaphragm initial pixel, σ_T is the standard deviation of the trachea final pixel, p_D is the average of the diaphragm initial pixel and p_T is the average of the trachea final pixel.

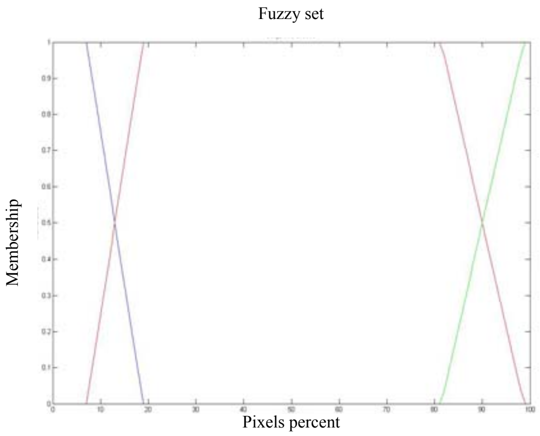


Fig. 2. Membership functions.

Another membership function is needed to evaluate if the maximum distance "D" and "I" found in the feature extraction is a correct measurement or not (if the measurement is part of the heart or part of another area like the diaphragm or trachea). The membership function *Similar*, indicates how much the maximum distance is similar to a maximum distance in the heart area (see Fig. 3). This fuzzy set will help to know if there is an abrupt growth of the measurements, meaning that the measurements pass from the heart area to the diaphragm area,

$$Similar = \begin{cases} -\frac{n}{10}+1, & if \quad 0 \le n \le 10 \\ 0, & otherwise \end{cases} \quad \textbf{(7)}$$

where *n* is the difference of pixels between the two measurements.

Using the *Similar* and *heart* membership functions the system is able to decide if the maximum distance is a measure of the heart or not. The fuzzy rule used to make the decision is presented as follows:

IF *p* IS *heart* OR *n* IS *Similar* THEN *m* IS correct **(8)**

where the OR connective used in the system is an algebraic sum of *Similar* and *heart*. The *heart* membership value is multiplied by 0.2 to lower the effect it has on the decision. In algebraic language this fuzzy rule is shown as,

$$correct(m) = \begin{cases} 1, & if \ m \ge 0.8 \\ 0, & otherwise \end{cases}, \quad \textbf{(9)}$$

where $m = \left(0.2 \cdot heart(p) + Similar(n)\right) - \left(0.2 \cdot heart(p) \cdot Similar(n)\right).$

With this rule the system knows if the measurement is correct, if it is not correct, then a new measurement should be taken and the system has to evaluate if the new measurement is correct or not, until it founds a correct measurement.

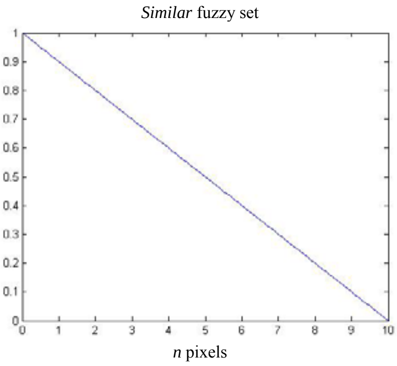


Fig. 3. Similar Membership function.

The problems found in the feature extraction of the heart size are solved using eq. (9) and the results are shown in Fig. 4, where the measurement obtained without implementing a fuzzy method (Fig. 4a) the right measurement of the heart is incorrect because it does not belong to the heart. After applying the fuzzy method (Fig. 4b) the system found the correct measurement of the right heart size.

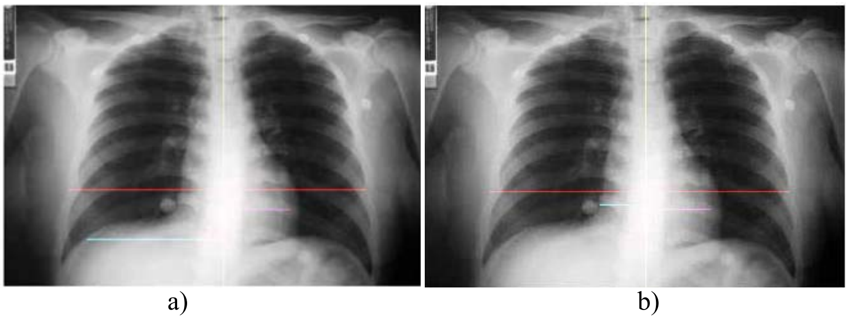


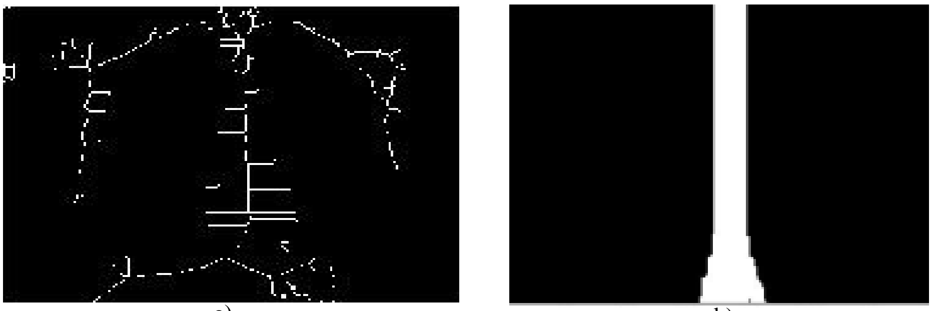
Fig. 4. Measurement of heart size: a) incorrect measurement, b) correct measurement.

4 Experimental Results

Several thorax radiographies where obtained from medical data bases. Here, the results to apply the mentioned algorithms to these images are presented. The features that are important in the thorax images are: size of the thorax, size of the heart, location of the mid line (trachea), distance between the mid line and the clavicles.

All the images were enhanced with the mentioned pre-processing algorithms (adjustment and histogram equalization) in order to obtain a similar contrast in all images according to the image that obtained the best results with the segmentation algorithms. Mathematical morphology algorithms were used to squeletize the pre-processed image. This process consists of several dilations and erosions; the new image is used to find the size of the thorax (see Fig. 5a). In order to find the mid line of the thorax, erosion is made with a rectangular structural element of size 5xm, where m is the vertical size of the image. The result of this erosion is shown in Fig. 5b.

In the images were used the edge detection algorithms. These new images (see Fig. 6) give us the necessary information to obtain the size of the heart (“D” and “I” defined previously).



a) b)

Fig. 5. Experimental results, a) Squeletize the image, b) Erosion.

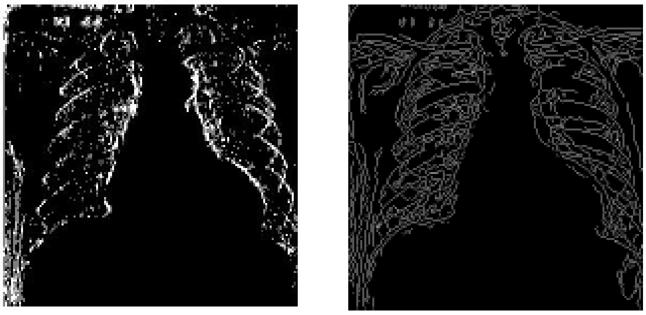


Fig. 6. Edge detection.

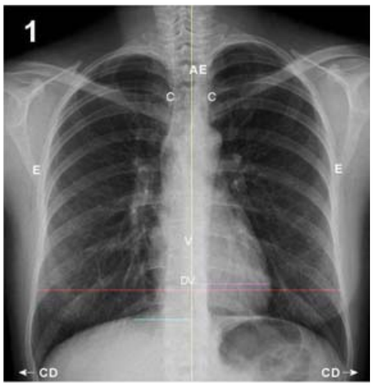


Fig. 7. Measurement of mid line and thorax size.

The measurements of the thorax size and mid line were achieved successfully. Fig. 7 shows the measurements.

Some problems were found to obtain the maximum size of the heart because there are an ambiguity regarding if the measurement is part of the heart, the trachea or the diaphragm. To solve this problem the implementation of a fuzzy system is needed. After making some measurements regarding the percentage of the image the heart belongs to, shown in the Table 1, some membership functions were created to represent if the maximum measurement belongs to the trachea, heart or diaphragm. In Table 1, (a), (b), and (c) are the number of vertical pixels in the image, start pixel of diaphragm, and percentage of pixels occupied by the diaphragm, respectively; (d) and (e) are the start pixel of heart, percentage of pixels occupied by the heart, respectively; (f), (g), (h), and (i) are the number of horizontal pixels in the image, the maximum distance found from the mid line of the thorax to a border found in the image, percentage of the distance, and mean distance in percentage, respectively.

Table 1. Measurements of image percents.

Image	a	b	c	d	e	f	g	h	i
1	205	161	21.46	25	66.34	226	43	19.02	11.94
2	418	401	4.06	103	71.29	599	69	11.51	8.68
3	170	154	9.41	20	78.82	341	46	13.48	8.5
4	181	-	0	29	83.97	326	40	12.26	9.2
5	175	143	18.28	20	70.28	303	39	12.87	9.57
6	247	226	8.5	34	77.73	394	50	12.69	8.88
7	135	127	5.92	13	84.44	218	31	14.22	10.09
8	321	-	0	71	77.88	640	138	21.56	13.9
9	194	153	21.13	27	64.94	300	59	19.66	10.66
10	289	236	18.33	23	73.70	432	105	24.30	16.89

Tables 2 and 3 present some experimental results in the case of use with and without fuzzy logic, respectively. The method used for comparison was using a fixed region in the image and take it as if this is the heart region, obtaining from this region the maximum distance as if this was the correct measurement of the heart size. The fuzzy method was implemented using the whole image and evaluating if the maximum distance found was the correct measurement of the heart. From these Tables, one can see that the proposed fuzzy method provides better results in terms of true positive and false negatives in comparison with the system without using fuzzy logic.

Table 2. Experimental results without fuzzy logic, where TP = true positive, FN = false negative, FP = false positive, and TN = true negative.

Image	IC Index by Program	Cardiomegaly by program	IC Index by manual measurement	Cardiomegaly by manual measurement	Difference
1	1.0000	Yes	0.45	No	FP
2	0.4182	No	0.3968	No	TN
3	0.6446	Yes	0.4020	No	FP
4	0.3533	No	0.3500	No	TN

5	0.6259	Yes	0.5957	Yes	TP
6	1.000	Yes	0.4320	No	FP
7	0.4471	No	0.4444	No	TN
8	0.7605	Yes	0.5714	Yes	TP
9	0.8855	Yes	0.5840	Yes	TP
10	0.3529	No	0.3500	No	TN
11	0.5369	Yes	0.3724	No	FP
12	0.6254	Yes	0.3917	No	FP
13	0.648	Yes	0.5526	Yes	TP
14	0.5787	Yes	0.3571	No	FP

Table 3. Experimental results with fuzzy logic.

Image	IC Index by Program	Cardiomegaly by program	IC Index by manual measurement	Cardiomegaly by manual measurement	Difference
1	0.4273	No	0.4500	No	TN
2	0.3798	No	0.3968	No	TN
3	0.396	No	0.4020	No	TN
4	0.3367	No	0.3500	No	TN
5	0.2937	No	0.5957	Yes	FN
6	0.3967	No	0.4320	No	TN
7	0.4087	No	0.4444	No	TN
8	0.5589	Yes	0.5714	Yes	TP
9	0.5954	Yes	0.5840	Yes	TP
10	0.3497	No	0.3500	No	TN
11	0.3833	No	0.3724	No	TN
12	0.3804	No	0.3917	No	TN
13	0.5302	Yes	0.5526	Yes	TP
14	0.414	No	0.3571	No	TN

To evaluate the performance of the proposed fuzzy method in terms of medical purposes, we compute the sensitivity and specificity. Sensitivity, is the probability that a medical test delivers a positive result when a group of patients with certain illness is under study, and specificity is the probability that a medical test delivers a negative result when a group of patients under study do not have certain illness,

$$Sn = TP / \left(TP + FN\right) \quad \textbf{(10)}$$

$$Sp = TN / \left(TN + FP\right) \quad \textbf{(11)}$$

where *Sn* is sensitivity, *TP* is the number of true positive that are correct, *FN* is the number of false negatives, that is, the negative results that are not correct, *Sp* is specificity, *TN* is the number of negative results that are correct and *FP* is the number of false positives, that is, the positive results that are not correct.

Table 4 shows the sensitivity, specificity, and error values obtained from the fuzzy method and the comparative method without fuzzy logic. We can observe that the specificity of the proposed method outperforms the comparative method. In the case of sensitivity, the proposed method has similar results in comparison with the comparative method.

Table 4. Comparative results in terms of sensitivity, specificity, and error values.

Method	Sensitivity	Specificity	Error
Comparative	75.00%	40%	42.85%
Fuzzy	75.00%	100%	7.14%

4 Conclusions

New diagnostic techniques are able to prevent and avoid deaths caused by cardiac insufficiencies. Analyzing thorax radiography by means of image processing is possible and can be used as an alternative diagnosis exam using the mentioned algorithms. This paper focus on finding cardiomegaly; but it is shown that there are other indicators that can be found in a thorax radiography that can help us make a better diagnosis. Fuzzy logic can be useful to achieve a good feature extraction of the radiographic image.

Acknowledgments. The work is supported by National Polytechnic Institute of Mexico.

References

1. Córdova Villalobos, J. A., Lee, G. M., Hernández Ávila, M., Aguilar Salinas, C. A, Barriguete-Meléndez, J. A., Kuri Morales, P., Lara Esqueda, A., Álvarez Lucas, C. H., Molina Cuevas, V., Barquera, S., Rosas Peralta, M., González Chávez, A.: Plan de Prevención Clínica de las Enfermedades Crónicas: sobrepeso, riesgo cardiovascular y diabetes mellitus 2007-2012 y Sistema de Indicadores de Diabetes en México. Comunicado de la secretaría de salud México. Revista Mexicana de Cardiología. Vol. 20, (1) pp. 42-45, (2009)
2. Giger D. K., MacMahon M., Nishikawa H., Schmidt R., Hoffmann R., Katsuragawa K., Sanada S., Behlen S., Sluis F. D.: Development of Digital Processing Techniques for Computer-Aided Diagnosis in Radiographic Images. In: The Third International Conference on Image Management and Communication in Patient Care. IMAC 93, pp. 110–115 (1993)
3. Jamroży M., Leyko T., Lewenstein K.: Early Detection of the Cardiac Insufficiency, Recent Advances in Mechatronics. Springer Berlin Heidelberg. pp. 407-411 (2010)
4. Bankman I.: Handbook of Medical Image Processing and Analysis. Vol. 1. Academic Press. Boston (2008)
5. Nixon M.: Feature Extraction & Image Processing. Academic Press. Boston (2008)

6. Gonzalez R.C.: Digital Image Processing using Matlab. Prentice Hall. New York (2003)
7. Ritter G.: Handbook of Computer Vision Algorithms in Image Algebra. CRC Press. Boca Raton-New York (2001)
8. Bezdek J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press. NewYork (1981)
9. Babuska R.: Fuzzy Modeling for Control. Kluwer Academic Publishers. Boston (1998)

Natural Language Processing

Language Models for Name Recognition in Spanish Spoken Dialogue Systems

German Tapia, Ivan V. Meza, and Luis Pineda

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS),
Universidad Nacional Autónoma de México,
Circuito Escolar s/n, Ciudad Universitaria, D.F., Mexico
tapia_g@uxmcc2.iimas.unam.mx,ivanvladmir@turing.iimas.unam.mx,
luis@leibniz.iimas.unam.mx
http://www.iimas.unam.mx/

Abstract. Current advances on dialogue system require the development of language models for automatic speech recognition that are not only domain or task specific but also sub-task specific (e.g. name, age or price recognition). This paper presents a method for the creation of language models for name recognition at the greeting stage of a conversation in spoken Spanish. In particular, we focus on the introductory phrases (e.g., *mi nombre es John/my name is John*). The method relies in the construction of a corpus for names, and we study two variants; in one names are uniformly distributed; in the other, names are represented in a manner consistent with their frequency in web-searches.

Keywords: Automatic Speech Recognition, Language Model, Spoken Dialogue System.

1 Introduction

Current practice permits the development of dialogues system (i.e., oriented to a specific task) with common underlying principles in a reasonable amount time [10, 4, 7]. However, Automatic Speech Recognition (ASR) technology is still a bottleneck for this kind of systems [14, 7]. There are several possible actions to improve the ASR module of a spoken dialogue system (e.g., to improve the acoustic model, to create better dictionaries, to use n-best list, etc.) In this paper, we focus on improving the language models. In particular, on the creation of language models for a specific conversational domain.

A large amount of research has been focused on the automatic creation of domain specific languages models [2, 3, 6, 15]. However, these models cannot be generalized to other domains, and much work and effort is required for developing new applications. To overcome this problem, we explore the automatic creation of language models which are sub-task specific and can be used in different applications and domains. A sub-task in a dialogue system is a small interaction between the user and the system which aims to resolve a specific conversational goal. Examples of subtasks are asking for the user's name or age, or providing

© G. Sidorov et al. (Eds.)
Advances in Soft Computing Algorithms
Research in Computing Science 49, 2010, pp. 145-154

the price of a product. In particular, we focus on the name recognition sub-task which happens during the introductory stage of a conversation. In this sub-task the user says phrases like *I'm called.../me llamo...* when his or her names is asked for. This is an important sub-task in a conversation, since the name can be used productively later on in the conversation for several purposes. However, it is hard to identify a good language model for such task, since there are a great amount of possible names, which have a unusual frequency distribution, with a few popular names, but the rest are not so common.

There are two general approaches for automatic language generation: using hand-coded grammars and an using out-of-domain corpus. In the former case, a set of rules and an in-domain lexicon are used to generate an in-domain corpus. This approach has the disadvantage that the quality of the corpus depends on the skill of the grammar's designer. Additionally, it is hard to make sure that the corpus frequencies correspond to the language used in common conversations. In the latter case, a large out-of-domain corpus is used to enrich a small in-domain corpus. This small in-domain corpus is usually called the seed. The result is a subset of the out-of-domain corpus but much larger than the seed. In this work, we propose to build a set of rules for the generation of name recognition expressions as hand-coded grammars approaches do. However, we implement two versions of this approach. The first version uses a uniform distribution of names, while the second is parametrized by the proportions of names in an out-of-domain corpus.

This paper is organized as follows. Section 2 reviews previous work. Section 3 presents our general methodology and its application to the name recognition sub-task. Section 4 describes the corpora used during the experiments. Section 5 describes the experiments performed and the results obtained. Finally, section 6 presents our conclusion and further work.

2 Previous Work

Previous work has focused on the creation of domain specific language models. Galescu et al. proposed two methods. The first uses a context free grammars to generate sentences in the domain. The second uses an out-of-domain corpus to identify the sentences which fit the domain best [3]. However, they results show that the interpolation of language models resulting of the two methods provides the best performance. Chung et al. presents a technique where an out-of-domain corpus is transformed to fit the domain [2]. This transformation is performed with two techniques. A set of hand-coded rules are used to rewrite the sentences, or automatic translation techniques are used to "translate" the sentences into the domain. Mahajan et al. uses information retrieval techniques to find an in-domain examples from an out-of-domain corpus [6].

These efforts focus on domain specific language models. That means that they aim to capture language specific to a domain, so that if a dialogue system is about tickets reservation the resulting language model using these techniques will contain information about tickets reservation. However, for a different dialogue system in another domain it will be necessary to create a different language model. In this work we look at language models which can be shared among dialogue systems; these will cover small parts of the dialogue but can be used in other systems and domains. These models could be either loaded dynamically for the specific part of the conversation or can be interpolated with other models to compose a final language model specific to the dialogue system.

A similar problem appears in directory assistance [1] and local business voice search [14] tasks. In these, a user asks to a spoken system the location of a business using its name. Business names also have an unusual frequency distribution, with a few words occurring very often and the rest, rarely. For both tasks, it has been identified a low performance for the ASR module and it has been proposed to use a set of approaches relying in the creation of language models. In this paper, we present a method for building language models too, but in our case we focus on name recognition

3 Sub-task Specific Language Model

Figure 1 shows a typical architecture of a dialogue system. There are three main stages: language understanding, dialogue management and language generation. The first stage assigns an interpretation or meaning to the input message. The purpose of the second is to choose the action to be taken next. Finally, the language generation stage renders a meaningful message directed to the user. The dialogue management component rests on a conversational protocol that models the task structure (e.g., [8]). This is, the conversation is divided into sub-tasks. For instance the ATIS systems [13], divides the reservation task into flight, car and hotel reservation. These tasks can be further divided into sub-tasks that identify or resolve a relevant piece of information, For instance, the identification of the destination city for the flight reservation task.

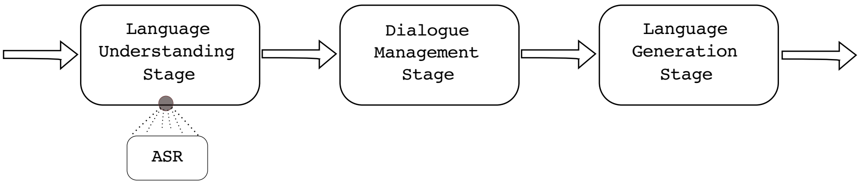

Language Understanding Stage
Dialogue Management Stage
Language Generation Stage
ASR

Fig. 1. Basic dialogue system architecture.

There are sub-tasks common to different systems. For instance, an introductory stage in which the user and the system introduce themselves to each other. This has as a further sub-task through which the user gives his or her name to the system. We called this sub-task *name recognition.* An instance of this is shown in Figure 2. The sub-task occurs between utterances 2 and 3. The name

No.	Speaker	Utterance
1	SYS	Hello, my name is Golem and this is the game Guess the card
2	SYS	There is a player. What is your name?
3	USR	Alba
4	SYS	Hello Alba, nice to meet you
5	SYS	How are you ?
...	...	...

Fig. 2. Excerpt of a introductory stage between a dialogue system (SYS) and user (USR).

recognition is important because the system can use the name to address the user later on, as in 4.

We have identified the name recognition sub-task in different dialogue systems: the Golem-2 system, a mobile robot which guides a poster session, asks for the user's name of the user [4]; the "Guess the card" system, which plays a game with children at a science museum also asks for the user's name [7]; the excerpt of the conversation in Figure 2 was taken from an actual conversation with this system. This sub-task occurs also in the *who-is-who* competition at *Robocup@home*[1] too. Here, a service robot looks for known and unknown participants within a scenario simulating a house. Once a participant is found the robot has to determine if the participant is known. If this is not the case, the robot has to introduce itself and ask for the participant's name, otherwise it has to address him or her by his or her name.

To address the problem of generation of sub-task specific language models we propose the mechanism showed on Algorithm 1. First a set of topics are identified. In the case of name recognition, there is only one topic involved: the *NAME*. For the step 2 a set of words belonging to the topic have to be defined. For the case of the *NAME* topic it is necessary to specify a list of names which will be identified. Additionally, we have to specify a set of templates on which such topics appear, like *my name is NAME*. In this template the *NAME* tag will be replaced by a name from the list. The next stage is to collect a set of statistics of the words conditioned on the topic. In this paper, we propose two strategies; the first consists of using an *uniform* distribution where all names appear with the same frequency. The second consists on using *web-searches hits* to approximate the distribution of the names. Once the corpus is available, a language model is generated using standard techniques. In particular, we explore the impact of using bigrams or trigrams for the language model.

[1] http://www.ai.rug.nl/robocupathome/

Algorithm 1 Language model generation.

```
1: Create a set of goal topics.
2: For each topic create a list of words belonging to such topic.
3: Create a set of templates which use such topics.
4: Obtain statistics of the words conditioned on the topic from an out-of-domain
   resource.
5: Generate sentences for the language model by substituting the words on the templates
   and by using the statistics of the words.
6: Generate a language model with the sentences generated in the previous words
   (using bigrams or trigrams).
```

4 Corpora

The algorithm 1 is used to generate a corpus until the step 5[2]. This corpus is then used in step 6 to generate a language model. We look into different strategies to generate this corpus. As mentioned two strategies are used to generate three corpora. All of these corpora uses the Spanish templates: *me llamo NAME/I'm called NAME*, *mi nombre es NAME/my name is NAME* and *NAME*. Recall that the*NAME* tag in the template will be replace by a name. We used a list of 624 names which were already available as a part of a functional dialogue system to generate $1,872$ sentences [7]. We also used *web-searches* hits of these 624 names to create a corpus. In this case, we repeated the phrases at least three times to match the distribution of names with the one of the web-searches. Table 1 shows the main properties of the three corpora. The first corpus uses the uniform strategy and we called it *small uniform*. However, this version of the corpora is small when compared with the third which uses web-searches hits. For these reason, we generated a *large uniform* by repeating each template $1,000$ times.

Table 1. Characterization of generated corpora.

	Small uniform	Large uniform	Web-searches
Word types	624	624	624
Sentences	1,872	1,872,000	2,875,884
Bigrams	2,474	2,474	2,474
Trigrams	4,324	4,324	4,324

We have also collected two audio corpora for testing the language models. The first consists of a collection of recordings of *final-users* interactions with a functioning dialogue system [7]. In particular, we identify the utterances in which the users utter their names. These audios have been manually transcribed with their orthographic transcription. However, this is a small corpus since we

[2] An implementation of this algorithm and the scripts used during the experiments can be found in the following link `http://code.google.com/p/language-models/downloads/`.

do not have that many instances of people saying their name to the system. On the other hand, we created an *artificial* generated corpus in which we ask ten users to tell ten unique random names in three different forms. With this setting we collected 300 instances of people uttering a name. Table 2 summarises the main propertied of these two corpora.

Table 2. Characterization of artificial and final-user corpora.

	Artificial corpus	Final-users corpus
Total participants	10	12
Total recordings	300	16
Total ways of uttering names	3	1
Total different names	100	12

Additionally, we use the DIMEx100 corpus to build the acoustic models used in the experiments [12]. The DIMEx100 corpus is a collection of 100 persons reading 50 different sentences and 10 common sentences in Spanish, phonetically transcribed. The corpus is phonetically balanced and has been used for the development of the Spanish ASR systems for the "Guess the card" and Golem systems. A speech recogniser developed with this corpus can be expected to have a performance of at least 48.3 Word Error Rate (WER) using a weak language model.

5 Experiments and Results

We generated six sub-task specific language models for name recognition using the three corpora described in section 4, with bigrams and trigrams. For testing we used the Sphinx 3 ASR system [5]. This setting allows us to measure the direct effect of the language models on the performance of the ASR module.

The first experiment consisted on measuring the WER of the six language models. We used the *artificial* generated corpus for testing. Additionally, we varied the language weight on the ASR module. This weight is used during decoding to couple the acoustic models with the language models [3]. The results are shown on Table 3. These results show better performance than previously reported [9]. However, it is important to keep in mind that the sentences on the *artificial* corpus are simpler. We also observe that the greater the weight the better the performance of the uniform models. This is not the case for the *web-searches* models which reach a high point at the weight of 7. However, we found that large language models, *large uniform* and *web-searches*, perform better small ones.

The WER gives a good measure of the ASR task; however it does not tell us much about the name recognition task. For this, we measured the performance

[3] We vary the weight from 5 to 13 which is the recommended range for Sphinx 3.

Table 3. WER for language models using *artificial* generated names corpus (the lower the better).

Language weight	Small Uniform		Large Uniform		Web-searches	
	Bigram	Trigram	Bigram	Trigram	Bigram	Trigram
5	35.1	35.9	26.0	26.6	26.1	27.1
7	29.8	30.7	25.6	24.9	**25.5**	**25.7**
9	28.2	28.7	25.3	25.4	25.6	25.7
11	27.1	27.4	25.6	25.4	25.8	25.8
13	**26.2**	**26.2**	**25.1**	**25.1**	26.4	26.4

of the language models on their capacity of recovering the right name and identifying the right context of the name words, this is the words which are or not a name. For this we defined two measures:*Name recovery* (*NR*) and *Name and context recovery* (*NCR*). *Name recovery* quantifies how many times the name was correctly retrieved from the hypothesis of the ASR system. *Name and context recovery*[4] quantifies how many times the right name and the context words were recovered from the recognition hypothesis; in this case we can calculate a precision, recall and $F1$-score. Table 4 presents the *name recovery* score and Table 5 the $F1 - score$ for *name and context recovery*. Similarly, as in the previous experiments, we show the results for different language weights.

Table 3 shows that the ASR performance was good. However, the *Name recovery* scores in Table 4 show that name recognition performance should be improved. The *uniform* language models outperform the *web-searches* ones. On the other hand, 5 shows that the *large* language models outperform the *small* ones; this measurement is closer to WER results showed in Table 3. We think that this change of better performance from uniform to large language models is due to the nature of the measurement. The WER and *Name and Context Recovery* scores are global measures. These consider the name recognition part, but they also consider the no-name words. On the other hand, *Name recovery* only considers the names. A large model will have better statistics about the context words than a small one. A uniform model will give any name a better chance to occur. This is consistent with the *large uniform* language model's good performance on both measurements. This is because it combines two important properties: uniformity and size.

In the second experiment, we evaluated the performance of the six language models using the *final-users* corpus. Table 6 presents a summary of the results. We use the better scores for the language weight and n-gram as shown in Table 3; we we used bigrams in all cases. Here, a pattern compatible with *artificial* corpus can be seen. They perform similarly for the WER score; however the *large uniform* corpus gives the best performance. For the name recovery score the

[4] Notice, that although we focus on the name recognition part, it is important to consider how good the approach is for the rest of the expressions, since they also could contain useful information.

Table 4. Name recovery scores for language models using *artificial* generated names corpus (the larger the better).

Language weight	Small Uniform		Large Uniform		Web-searches	
	Bigram	Trigram	Bigram	Trigram	Bigram	Trigram
5	33.6	32.3	34.0	34.0	**33.0**	**33.0**
7	34.0	33.6	**34.3**	**34.3**	32.6	32.6
9	34.0	34.0	34.0	34.0	30.3	30.3
11	34.3	34.3	33.3	33.3	29.3	29.3
13	**34.3**	**34.3**	32.6	32.6	27.6	27.6

Table 5. Name and context recovery $F1$-scores for language models using *artificial* generated names corpus (the larger the better).

Language weight	Small Uniform		Large Uniform		Web-searches	
	Bigram	Trigram	Bigram	Trigram	Bigram	Trigram
5	58.7	57.8	71.11	70.83	71.0	70.1
7	64.1	64.0	72.61	72.34	**71.5**	**71.4**
9	66.4	66.2	71.93	72.07	71.0	71.0
11	68.4	68.3	**72.34**	**72.34**	70.7	70.7
13	**69.6**	**69.6**	71.93	71.93	69.9	69.9

small uniform language perform better than the others. However, for the more global score, NCR, the situation is reversed and the larger corpora outperforms the smaller.

Table 6. Scores for language models using *final-user* corpus (WER, the smaller the better; NR and NCR, the larger the better).

Language models	WER	NR	NCR
Small uniform	39.6	**31.3**	43.9
Large uniform	**37.5**	25.0	**51.2**
Web-searches	39.6	25.0	**51.2**

6 Conclusion and Future Work

In this paper we presented a method to generate sub-task specific dialogue models. The difference of this work with previous approaches is that it builds small language models which can be shared among different dialogue systems which have the same sub-task. In particular, we focus on the name recognition sub-task. This sub-task is part of an introductory stage in conversation. In this sub-task the user gives his or her name to the system. It is important to identify the name since the system can address the user by his or her name later on during the conversation. This sub-task is common among different dialogue systems in which to know the name of the user is required. On the other hand, names are challenging to model since there is a large amount of names and their frequency distribution is unusual, with few names with high frequency and the rest with extremely low frequencies.

We explored two strategies to generate three corpora for the language models. The first consisted in using a uniform distribution for the names. The second consisted in using an approximate distribution for the names based on web-searches hits. We generated three corpora, a *small uniform*, a *large uniform* and *web-searches*. With these three corpora we generated six language model, using bigrams and trigrams for each of the three corpora. These six language models allows us to measure the effect of the strategy on the ASR module.

For testing two corpora were collected. The first is an *artificial* corpus in which 10 subject utter 10 names in three different ways. The second is a *final-user* corpus in which an actual users utter their name to a dialogue system. We tested the ASR performance using the different language models in three different ways. We first used the standard Word Error Rate (WER). The second consisted on identifying how many names could be recovered from the ASR hypothesis (i.e., Name recovery, *NR*). Finally, the third measurement consisted on comparing the times the names could be recovered and the times the context words could be recovered (i.e., Name and context recovery, *NCR*). The main difference between the two last measurements is that the first is local to the recognition of the names, and second is global to the task of recovering information.

Our experiments showed that the combination of a large corpus using an uniform distribution provided the best balance for the language model. In particular, we obtained good local (*NR*) and global scores (WER and *NCR*). The *web-searches* corpora was not good to capture a good performance when trying a local score as the *NR*. Another important result was to discard the trigrams for futures implementations of the language models, since the best performance is reachable using the less complex bigrams. This is consistent with our intuition about the name recognition sub-task on which we rarely find more than four words for the involved phrases (e.g., *mi nombre es NAME*/my name is).

We have also identified points of further improvement and new research questions. We explored two strategies to generate the corpora for the language models so far. However, we could get a better approximation of the names by using other source of information, for instance name statistics from schools or the national registry. We are also looking into complement our *final-users* corpus with a larger amount of examples of people uttering their names in functioning dialogue system. We are also looking into generating language models for other sub-tasks like age identification. We also need to test these language models in a working dialogue system. We plan to test the dynamic option, in which a specific language model is loaded for a particular part of the conversation, and the interpolation option, in which a set of language models are interpolated to create a language model specific to a dialogue system.

Acknowledgements. We thank the enthusiastic support of the members of the DIME and Golem group at IIMAS and at Science Museum Universum-UNAM. We also gratefully thank the support of grants CONACYT 81965 and PAPPIT-UNAM IN-104408 and IN-115710.

References

1. Buntschuh, B., Kamm, C., Di Fabbrizio, G., Abella, A., Mohri, M., Narayanan, S., Zeljkovic, I., Sharp, R., Wrigth, J., Marcus, S., Shaffer, J., Duncan, R., Wilpon, J.: VPQ: a spoken language interface to large scale directory information. In: Proceedings of ICSLP (1998)
2. Chung, G., Seneff, S., Wang., C.: Automatic Induction of Language Model Data for a Spoken Dialogue System. In: Proc. SIGdial-05 (2005)
3. Galescu, L., Ringger, E., Allen, J.: Rapid Language Model Development for New Task Domains. In: Proc. LREC, Granada (1998)
4. Avilés, H., Alvarado-González, M., Venegas, E., Rascón, C., Meza, I., Pineda, L. A.: Development of a Tour–Guide Robot Using Dialogue Models and a Cognitive Architecture. In: Proceedings of Iberamia 2010, LNAI (2010)
5. Huerta, J. M., Chen, S.J., Stern R. M.: The 1998 carnegie mellon university sphinx-3 spanish broadcast news transcription system. In: Proc. of the DARPA Broadcast News Transcription and Understanding Workshop (1999)
6. Mahajan, M., Beeferman, D., Huang, D.: Improved Topic-Dependent Language Modeling Using Information Retrieval Techniques. In: Proc. ICASSP, pp. 541-544 (1999)
7. Meza, I., Salinas, L., Venegas, E., Castellanos4, H., Chavarría, A., Pineda L. A.: Specification and Evaluation of a Spanish Conversational System Using Dialogue Models. In: Proceedings of Iberamia 2010 (2010)
8. Pineda, L. A.: Specification and Interpretation of Multimodal Dialogue Models for Human-Robot Interaction. In: G. Sidorov (Ed.) Artificial Intelligence for Humans: Service Robots and Social Modeling, SMIA, México, pp. 3350 (2008)
9. Pineda, L. A., Castellanos, H., Cuétara, J., Galescu, L., Juárez, J., Llisterri, J., Pérez-Pavón, P., Villaseñor, L.: The Corpus DIMEx100: Transcription and Evaluation. Language Resources and Evaluation (2009)
10. Pineda, L. A., Meza, I., Salinas, L.: Dialogue Model Specification and Interpretation for Intelligent Multimodal HCI. In: Proceedings of Iberamia 2010, LNAI (2010)
11. Oparing, I.: Statistical Language Models for Dialogue Systems. Technical Report (2006)
12. Pineda, L. A., Castellanos, H., Cuétara, J., Galescu, L., Juárez, J., Llisterri, J., Pérez-Pavón, P., Villaseñor, L.: The Corpus DIMEx100: Transcription and Evaluation. Language Resources and Evaluation (2009)
13. Deborah A. Dahl et al.: ATIS 3 training data. Linguistic Data Consortium, Philadelphia (1994)
14. Stent, A., Zeljković, I., Caseiro, D., Wilpon, J.: Geo-Centric Language Models for Local Business Voice Search. In: Proceedings of HLT/NAACL-2009 (2009)
15. Wan, V., Hain, T.: Strategies for language model web-data collection. In: Proc. ICASSP, UK (2006)

Translation of Spanish Statistics Expressions to SQL

Ismael Esquivel Gámez, Ma. de los Ángeles Marrujo, and Omar Pérez García

Universidad Veracruzana,
Mexico
iesquivel@uv.mx, angeles_marrujo@hotmail.com, omarpg85@hotmail.com

Abstract. Grouping and aggregate data to obtain summarized values from a database is one of the most representative ways of generating information through a computer system. End users when requesting reports of this nature incorporate statistical expressions as *total, subtotal, suma, promedio, máximo, mínimo, etc.* (total, subtotal, sum, average, maximum, minimum, less, more, etc.), like in: ¿Cuál es el total de salarios pagados a los diseñadores? (What is the total wages paid to designers?). This paper presents a system for recognizing the statistics expressions in Spanish, found in natural language queries made to a database and converting them to SQL. It is a module of a natural language interface to database (NLIDB), still developing. The system handles three types of typical Spanish queries involving statistical expressions.

Keywords: Natural Language Interfaces to Data Based, SQL, Statistical Expressions.

1 Introduction

The natural language interfaces to databases (NLIDBs) are systems that allow end users to access a database introducing their queries in natural language [1]. Translate those queries to a database language like SQL, run the command and show the results. Some of these requests contain statistical expressions equivalent to processing the data stored in one or more tables, through the aggregated and grouping.

With SQL [2], data can be grouped and aggregated so that users can interact with them at a higher level of granularity that as they ar e stored in the database. The aggregated is carried out [3], by internal functions of SQL that allow you to calculate a single value from the rows in a result set or from the values returned by a arithmetic expression.

Although the commercial database engines support a wide of internal functions, in this paper we handle the functions of the ANSI SQL-92 standard, to extend the usefulness of the prototype presented.

In a natural language query may appear multiple statistical expressions, creating some degree of complexity in its conversion. In this report, we describe a module of a Spanish NLIDB, named SNL2SQL, under construction. Converts the query issued by an end user, containing statistical expressions, to the corresponding SQL SELECT command.

2 Related Work

According to the classification of queries in [4], consisting of six types, this paper focuses on solving the query type that requires special functions (number 5). For this reason, works in Spanish and English that translate this type to SQL, were reviewed.
In [5] handles a module, which translates from Spanish to an SQL query through four steps. In each, the query is subjected to several transformations by: The generation of an equivalent template semantic, the adaptation of the template data to those found in the database, the generation of the complete and final template and finally, the formulation of the corresponding SQL statement. In their work, the use of internal and external functions, it was proposed. For the first ones: Unidades, diferencia_fecha, superlativo, cuenta y promedio (Units, date difference, superlative, average and count), of which only the last three were translated directly into SQL aggregate functions and the other required a pre-conditioning.

To improve the process of setting up their NLIDB [6], they propose ontology to achieve greater acceptance and therefore, accessibility. The ontology defines categories to organize the concepts that have the context of the database. Among the highest hierarchy, were defined: Database elements, words, sets of synonyms (synsets) and functions. The latter were classified in: SQL Functions, User defined and call-link. For the former, groups of words or synsets, were equivalent to aggregate functions, semantically speaking.

In their methodology [7], they used natural language generation (NLG) as an alternative to remove the interpretation stage of queries. Those are constructed by the user through successive interactions with a natural language text, called feedback text. Changes to it are reflected in the semantic content of the query and are made in parts of the text enclosed by [], named anchors. In these, values can be changed, add aggregate functions, change or remove filters and order conditions; to generate the corresponding SELECT statement.

In their case [8] designed a graphical user interface and a grammar-driven phrases, like natural language grammar, called PDG. To enter a query and displaying the results, it follows a process of constructing sentences, targeted for users with no experience in data access and visualization. PDG manages aggregate functions with operators such as “number of”, “sum of”, and “avg of”; and whose argument is specified by [data], which can be used with "For Each data" phrase. The lowest and larger operators are used to obtain minimum and maximum. It also handles grouping facilities to generate queries shorter and clearer than SQL.

3 Proposed Approach

As part of the implementation and after analyzing the corpus of queries made to the test database by students and faculty from IT area, three types of questions associated with statistical expressions were found. Table 1 shows the types that this proposal addresses and the SELECT commands associated.

Table 1. Types of queries and associated commands.

Type	Query	SELECT command
1	*¿Cuál es el total de salarios?* *(What is the total wages?)*	Select sum(Empl.salary) From Empl
2	*¿Cuántos empleados se tienen por cada departamento?* *(How many employees have every department?)*	Select Empl.Workdept, count(*) From Empl Group by 1 Order by 2 Desc
3	*¿Quién gana más salario de los empleados?* *(Who earns more salary for employees?)*	Select Empl.Empno, Sum (Empl.salary) From Empl Group by 1 Order by 2 Desc

Although the third type can be solved with a subquery, we have found advantages with the approach proposed, among some, the orderly deployment of multiple records, allowing a more comfortable view of the information collected.

Likewise, were detected most common statistical expressions found in queries in Spanish, which are partially showed in Table 2. After a comprehensive analysis of questions and expressions, it was found that some of them created uncertainty, about the type of question that was being dealt (1 or 3), being regularly the following: *más, mayor, menos, menor* (*more*, *higher*, *less*, *smallest*), named special terms.

3.1 Assumptions and Limitations

For the database, we used the assumptions that were proposed in [9], from C1 to C15 and for the queries made by the end user (A1 and A2), adding the management of mandatory queries ("Displays the maximum salary").

Managing expressions as an argument to a function is limited to those explicitly handle the attributes and processing, for example *"El total del salario menos la commission"* (*The total salary minus the commission*).

This first prototype does not handle the filtering of groups with the HAVING clause of the SELECT or relationship between tables (JOIN).

3.2 Previous Configuration

Before using the module, the administrator needs to do certain tasks, among which are:

3.2.1 Registration of Domain Values. For attributes that handle a set of valid values that can take (domains), mainly for those with short values, for example SEX ('F', 'M'), is necessary to capture the synonyms for these values: 'MUJERES',

’FÉMINAS’, ’FEMENINO’, ’HOMBRES’, ’VARONES’, ’MASCULINO’, etc. ('WOMEN', 'FEMINIST', 'FEMALE', 'MALE', 'MALE', 'MALE', etc.), in a file called Content Dictionary. It serves to facilitate processing, primarily the WHERE clause and thus, it encourages the use of terms, characteristic of the enterprise environment, where the prototype is implemented.

Table 2. Statistical Expression and aggregate functions.

Expressions	Data type	SQL Function
Total, suma, sumatoria, subtotal, … (Total, sum, sum, subtotal…)	Numeric	SUM([DISTINCT] X)
Promedio, media, … (Average, average,…)	Numeric	AVG([DISTINCT] X)
Menos, menor, mínimo, más (reciente), …, (Less, smallest, minimum, Most (recent)…,)	All	MIN([DISTINCT] X)
Más, mayor, máximo, menos (viejo), …, (More, higher, maximum, Less (older), …,)	All	MAX([DISTINCT] X)
Cuántos, cantidad, número de, cuenta de,…, (How many, quantity, number of…, count of…,)	Values	COUNT([DISTINCT] X)
Cuántos, cantidad, número de, cuenta de,…, (How many, quantity, number of…, count of…,)	Tuples	COUNT(*)

3.2.2 Definition of Grouping Attributes. It is required to select the attributes that are grouped in a dictionary. Also a series of queries of the database can be run for purposes of automating the operation. For each attribute, is issued a command like this:

Select <table.attribute>, count(*) from <table> group by 1 having count(*) > 1

3.2.3 Election of the Attribute to Use by Default. It is necessary to define for each table, a unique attribute that fully describe each tuple and that this is enough descriptive, like in [10], when the query does not express the grouping attribute, for example *¿Quién gana más salario?* (*Who earns more salary?*). For the present case, were chosen: Employee Number (Empl.Empno), department name (Dept.Deptname), project name (Project.Projname).

The last two definitions are stored in the domain dictionary, which is described below.

The prototype uses two dictionaries, the main also called "domain" and the secondary, called "content." The first was built from a synonym dictionary and the database metadata, based on the paper of Pazos *et. al* [8]. From the metadata is extracted by each table: name, description, and details of their columns (name, data type, size, permissibility of null values and its description. From the processing of the latter, are obtained nouns and partnership to the columns and tables, that in its description containing them or their synonyms. Also is obtained the equations representing the links between the tables to process queries with join. The dictionary content is updated similarly for columns that handle sets of valid values.

For this work, the synonym dictionary has been built manually using the online dictionary provided in [11], but the idea is to generate it automatically, in the future.

4 Main Algorithm

The translation process is carried out following the next steps:

For all three types of questions:

1. Statistical expressions are identified in the question in any of the following cases:
   a. There are one or more and do not correspond to the special terms, in which case proceed to Step 2. For example the queries "*¿Cuál es el monto total de salarios para diseñadores y operadores?*" (*What is the total amount of salaries for designers and operators?* or "*¿Cuál es el salario máximo y salario mínimo de los gerentes?*" (*What is the maximum wage and minimum wage for managers?*).
   b. There are two or more and one of them is special term like in "*¿Qué departamento tiene el promedio de salarios más alto?*" (*What department has the highest average salary?*), in which case, the expression is ignored momentarily and move on to step 2
   c. There is one and belongs to such terms, for which we continue to step 8. For example, "*¿Qué mujeres ganan más salario?*" (*What women earn more salary?*).
2. Found expressions define aggregate functions to use according the Table 1, eliminating such expressions of the query.

3. Determine the argument of the function, from the processing of words associated with the statistical expression, arranged in one of the following cases:
    a. Next, before or after, like in *"Salario mínimo"* (*minimum wage*).
    b. Connected with the preposition "*de*" (of) or "*de*" (of) and an article. Examples: *"Total de bonos"* (*Total of bonus*), "*promedio de la commission*" (*Average of the commission*).
    c. As a noun attached to "Cuántos" (How many), for example: *"¿Cuántos empleados…?"* (*How many employees…?*).
    d. As an arithmetic expression, denoted by the use of preposition (*"of"*) and conjunction (*"and"*), as resolved in [9]. An expression like *"El total de salarios y comisión"* (*Total of wages and commission*).
    e. Connected to words that involve the processing of non-repeated values, for example *"Únicos"*, *"distintos"*, *"diferentes"* (*Unique, different, etc.*), in which case the term DISTINCT is added to the argument as a prefix.

    Then it searches the domain dictionary to determine the pair table-attribute that will serve as an argument. The question is marked as "invalid" when it is not found or when the attribute handle a data type incompatible with the function. When there is more than one pair, the query is marked as "ambiguous". In the case of the COUNT function, the argument is the symbol (*), when asked the number of rows, therefore only needs to extracted from the question the name of the table to process, like in *"¿Cuántos trabajadores son varones?"* (*How many workers are male?*).
4. Nouns for the argument of the function and eventually, the terms of distinction, are removed from the question.

Question type 2:

5. All expressions involving grouping are identified, for example *"Por"*, *"por cada"*, *"en cada"*, *"en"*, *etc.* (*By, for every, every, in, etc*.), in the order they appear in the question.
6. Nouns associated with the previous expressions, are sought in the domain dictionary entries, only for those attributes previously marked as grouped, to determine the column(s) to use in the grouping clause. If not found, the question is marked as "invalid" and if there is more than one, as "ambiguous" Then both, nouns and expressions associated, are removed from the question.
7. Add the "order by" clause, indicating the numbers of columns that functions occupy in the SELECT clause with the DESCENDING option, to show higher figures first.

Question type 3:

8. Given that there has not been defined the aggregate function, nouns associated with the special terms: *más, mayor, menos, menor* (*more, higher, less, smallest*) are processed as in step 3, to get the argument of the function, which will be one of the following:
    a. If the data type is numeric, SUM(X)
    b. If it is date or date and time, MIN(X)
9. It is determined the column(s) used (s) in the grouping clause, searching for the remaining nouns in the dictionary domain. If not found, for example: *¿Quién gana más salario de los empleados?* (*Who earns more salary for employees*?), the default defined column, described previously, is used. Then the *"order by"* clause is added, indicating the number of the column that the function occupies in the select clause, adding the Ascending or Descending option, according to the special term yet found in the text of the question. So Ascending option is associated with terms such as *menos*, *menor* (*less, smallest*), and DESCENDING with *más, mayor* (*more, highest*). However, there are nouns like *old* and *antique*, in which case, result in reverse.

Finally, for the three types:

10. Expressions of filtering results are processed, to build the WHERE clause, then removing those expressions.

5 Experimental Results

The test database used is shown in Figure 1. The corpus of natural language queries was formed after the presentation of the Database and explanation of the project to teachers and students of the IT area. They were asked to write queries of the three types, and 27, 17 and 21, respectively were obtained, after eliminating similar.

From the corpus generated, which consisted of 65 requests, the results obtained are shown in Table 3. The conversion process took questions from a file and marked the non-translated queries with an "I" for invalid, "A" for ambiguous, and "X" for incorrect.

It was found that all the "invalid" queries, contained terms that could serve as function argument, but not explicitly indicated attributes and processing. For example: *"¿Cuál es la edad promedio de los gerentes?"* (*What is the average age of managers?*), *"¿Cuál ha sido la duración promedio de los proyectos?"* (*What was the average duration of projects?*). The "Incorrect" were those that contained no statistical expression, like in *"¿Cuál es el bono anual que reciben los diseñadores?"* (*What is the annual bonus received by designers?*)

In general, the results were quite promising for being a first approximation, and the support for implicit arithmetic expressions, undoubtedly will improve the effectiveness. It is important to note that verification of the translation was done manually, because no connection has yet implemented with a commercial database engine.

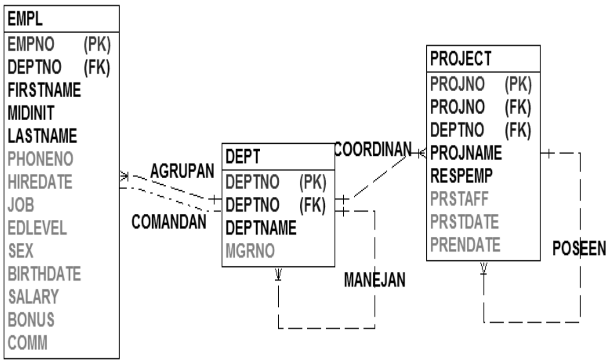


Fig. 1. Logical diagram of the test database.

Table 3. Obtained results.

Query type	1	2	3	Total
Queries Formulated	27	17	21	65
“Invalid”	3	1	2	6
“Incorrect”	2	3	4	9
“Ambiguous”	0	0	0	0
Translated	22	13	15	50
Effectiveness	81.48%	76.47%	71.43%	76.46%

The programming language used for development was Rexx (REstructured eXtended eXecutor), a programming language developed at IBM by Michael Cowlishaw, of which there are numerous open source implementations available [12]. Version 3.4 of the Regina Rexx interpreter for Windows was used and was chosen because it has a wide set of functions, especially for the treatment of texts and dates. Another crucial advantage to use Rexx is that it is a multiplatform language. It can, with slight modifications, run on any computer under any operating system and work.

5 Conclusions

The module presents a very specific approach to treat statistical expressions that are translated into SQL functions, grouping and order clauses. It is contemplated to test it with real-applications databases containing a larger number of attributes of grouping and aggregated.

The structure of queries from the corpus formed has been analyzed and found three types that predominate in reporting requirements. Such questions are representative of the reports requested by the executives and middle management, and therefore, automatically translating is a key factor in development of the SNL2SQL NLIDB. However, improvements are needed. One of the areas to improve is carry out the translation of more complex queries containing expressions of group filtering and the relationship between two or more tables. Another of the improvements is the automation of both, the generation of the Dictionary of Synonyms and the verification of the conversion.

There were several queries that were related to variables not considered in the database, but with a more refined processing, can be obtained. It is possible to enrich the tasks of the previous configuration and allow the entry of these variables, the associated database attributes, system variables and arithmetic operations. For that reason, the development of an enhanced version, to improve effectiveness, is a priority line of work.

Acknowledgement. Work supported by the Mexican Government (SEP/PROMEP), grant 103.5/09/4482.

References

1. Androutsopoulos, I., Ritchie, G., Thanisch, P.: Natural language interfaces to databases-An introduction. Journal of Language Engineering, 1(1) pp.29-81 (1995)
2. Beaulieu, A.: Learning SQL. Second Edition, O'Reilly Media, Sebastopol (2009)
3. Viescas, J.L., Hernandez M.J.: SQL Queries for Mere Mortals: A Hands-On Guide to Data Manipulation in SQL. Second Edition. Pearson Education. Massachusetts (2008)
4. González, J., Pazos, R. A., Gelbukh, A., Sidorov, G., Fraire, H., Cruz, C.: Prepositions and conjunctions in a natural language interfaces to databases. In: Thulasiraman, P., et al. (eds.) ISPA 2007 Workshops. LNCS Vol. 4723, Springer, Heidelberg, pp. 173-182 (2007)
5. Colás, J.: Estrategias de incorporación de conocimiento sintáctico y semántico en sistemas de comprensión del habla continua en español. Estudios de lingüística española (ELIEs). Publicación periódica de monografías sobre lingüística española. http://elies.rediris.es/elies12/index.html (2001)
6. Zarate, J. M., Pazos, R. A., Gelbukh, A., Pérez J.: Improving the customization of natural language interface to databases using an ontology. In: Gervasi, O, Gavrilova, M. (eds.) ICCSA 2007. LNCS Vol. 4705. Part. I, pp. 424-435 (2007)
7. Hallett, C., Hardcastle, D.: Towards a bootstrapping NLIDB system. In: Kapetanios, E., Sugumaran. M. Spiliopoulou (eds.) NLDB 2008. LNCS, Vol. 5039, Springer, Heidelberg, pp. 199-204 (2008)
8. Lee, S. Y., Neumann, U.: A Phrase-Driven Grammar System for Interactive Data Visualization. In: Börner, K., Gröhn, M., Park, J., Roberts J. C. (eds.) Visualization and Data Analysis. SPIE Int. Soc. Opt. Eng. Vol. 6829 (2008)
9. Pazos R., Pérez J., González J. J., Gelbukh A., Sidorov G. y Rodríguez M.: A Domain Independent Natural Language Interface to Databases Capable of Processing Complex Queries. MICAI 2005: Advances in Artificial Intelligence. MICAI 2005: pp.833-842 (2005)

10. Stratica, N., Kosseim, L., Desai, B.C.: Using semantic templates for a natural language interface to the CINDI virtual library. Data & Knowledge Engineering. Vol.55 No.1. pp. 4-19 (2005)
11. Rodríguez, S., Carretero J:. COES: Herramientas para Procesamiento de Lenguaje Natural en Español. http://www.datsi.fi.upm.es/~coes/interactivo/sinonimos.cgi (2008)
12. Mertz, D.: Rexx for everyone: Scripting with Free Software Rexx implementations. http://www.ibm.com/developerworks/library/l-rexx.html (2004)

Author Index

Índice de autores

Editorial Board of the Volume

Comité editorial de volumen

Jean-Bernard Hayet
Sergio Hernandez
Arturo Hernández
Hugo Hidalgo
Larry Holder
Joel Huegel
Marc-Philippe Huget
Seth Hutchinson
Dieter Hutter
Pablo H. Ibarguengoytia
Héctor Jiménez Salazar
Moa Johansson
Young Hoon Joo
Chia-Feng Juang
Vicente Julian
Hiroharu Kawanaka
Mario Koeppen
Mark Kon
Vladik Kreinovich
Ricardo Landa-Becerra
Reinhard Langmann
Yulia Ledeneva
Yoel Ledo Mezquita
Chang-Yong Lee
Guillermo Leguizamón
Eugene Levner
Tao Li
James Little
Giovanni Lizárraga Lizárraga
Aurelio Lopez
Edgar Lopez
Francisco Luna
Gabriel Luque
Rene Mac Kinney
Tanja Magoc
Jacek Malec
Luis Ernesto Mancilla Espinosa
Claudia Manfredi
Jose Marroquin
Ricardo Martinez
José Fco. Martínez-Trinidad
Alfonso Medina Urrea
Patricia Melin
Efrén Mezura-Montes
Mikhail Mikhailov
Gabriela Minetti
Dunja Mladenic
Francois Modave
Raul Monroy
Manuel Montes-y-Gómez
Oscar Montiel
Eduardo Morales
Rafael Morales
Guillermo Morales-Luna
Jaime Mora-Vargas
Angel E. Munoz Zavala
Masaki Murata
Tomoharu Nakashima
Juan Antonio Navarro Perez
Antonio Nebro
Atul Negi
Juan Carlos Nieves
Juan Arturo Nolazco Flores
Alberto Ochoa Zezzatti
Ivan Olmos
Constantin Orasan
Magdalena Ortiz
Mauricio Osorio
Daniel Pandolfi
Ted Pedersen
Alejandro Peña Ayala
Arturo Perez
David Pinto
Michele Piunti
Silvia Poles
Eunice E. Ponce-de-Leon
Edgar Alfredo Portilla-Flores
Pilar Pozos
Jorge Adolfo Ramirez Uresti
Alonso Ramirez-Manzanares
Zbigniew Ras
Fuji Ren
Orion Fausto Reyes-Galaviz
Carlos A Reyes-Garcia
María Cristina Riff
Mariano Rivera
Eduardo Rodriguez
Leandro Fermín Rojas Peña
Paolo Rosso
Jianhua Ruan
Salvador Ruiz Correa
Carolina Salto
Gildardo Sanchez
Frank-Michael Schleif

Roberto Sepulveda
Leonid Sheremetov
Grigori Sidorov
Gerardo Sierra
Thamar Solorio
Humberto Sossa Azuela
Graham Steel
Luis Enrique Sucar
Javier Tejada Cárcamo
Hugo Terashima
Sulema Torres Ramos
Gregorio Toscano-Pulido
Fevrier Valdez
Aida Valls
Berend Jan van der Zwaag
Maria Vargas-Vera
Karin Verspoor
Francois Vialatte
Javier Vigueras
Eliseo Vilalta
Manuel Vilares Ferro
Andrea Villagra
Miguel Villarreal
Thomas Villmann
Toby Walsh
Julio Zamora
Carlos Mario Zapata Jaramillo
Ramon Zatarain
Claudia Zepeda Cortes
Qiangfu Zhao

Additional Reviewers

Árbitros adicionales

Rita M. Acéves-Pérez
Esteve Almirall
Tristan Behrens
Federico Bergenti
Janez Brank
Nils Bulling
Noe Alejandro Castro Sánchez
Elva Díaz
Gibran Etcheverry
Ivan Figueroa
Jon Ander Gómez
Maria Auxilio Medina
Sabino Miranda
J. Arturo Olvera-López
Santiago Ontañón
John Quarles
Daniel Ramirez-Cano
Jorge Alberto Soria-Alcaraz
Ivan Varzinczak
Esaú Villatoro-Tello
Victor Manuel Zamudio Rodriguez

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Noviembre de 2010
Printing 500 / Edición 500 ejemplares