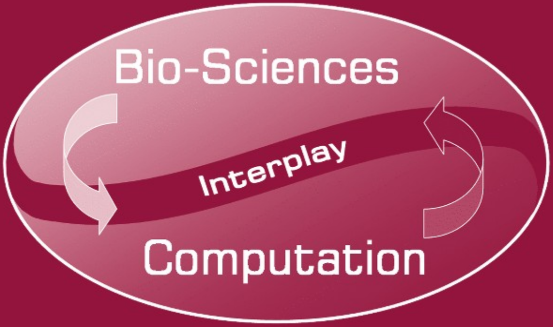
José Manuel Ferrández
José Ramón Álvarez Sánchez
Félix de la Paz
F. Javier Toledo (Eds.)

LNCS 6686

Foundations on Natural and Artificial Computation

4th International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2011
La Palma, Canary Islands, Spain, May/June 2011, Proceedings, Part I

1 Part I



Springer

Lecture Notes in Computer Science 6686

José Manuel Ferrández
José Ramón Álvarez Sánchez
Félix de la Paz
F. Javier Toledo (Eds.)

Foundations on Natural and Artificial Computation

4th International Work-Conference on the Interplay
Between Natural and Artificial Computation, IWINAC 2011
La Palma, Canary Islands, Spain, May 30 - June 3, 2011
Proceedings, Part I

Springer

Volume Editors

José Manuel Ferrández
F. Javier Toledo
Universidad Politécnica de Cartagena
Departamento de Electrónica
Tecnología de Computadoras y Proyectos
Pl. Hospital, 1
30201 Cartagena, Spain
E-mail: info@iwinac.org

José Ramón Álvarez Sánchez
Félix de la Paz
Universidad Nacional de Educación a Distancia
E.T.S. de Ingeniería Informática
Departamento de Inteligencia Artificial
Juan del Rosal, 16, 28040 Madrid, Spain
E-mail: info@iwinac.org

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-21343-4 e-ISBN 978-3-642-21344-1
DOI 10.1007/978-3-642-21344-1
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: Applied for

CR Subject Classification (1998): F.1, F.2, I.2, G.2, I.4, I.5, J.3-4, J.1

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Searching for the Interplay between Natural and Artificial Computation

The general aim of these volumes, continuing with ideas from Professor José Mira and with neurocybernetic concepts from Wiener and W.S. McCulloch, is to present a wider and more comprehensive view of the computational paradigm (CP), proposed by Alan Turing, than usual in computer science and artificial intelligence (AI) and to propose a way of using that which makes it possible: (1) to help neuroscience and cognitive science, by explaining the latter as a result of the former, (2) to establish an interaction framework between natural system computation by posing a series of appropriate questions in both directions of the interaction, from artificial systems to natural systems (in computational neuroscience), and from natural systems to artificial systems (in bioinspired computation). This is the main motivation of the International Work-conference on the Interplay between Natural and Artificial Computation, trying to contribute to both directions of the interplay:

I: From Artificial to Natural Computation. What can computation, artificial intelligence (AI) and knowledge engineering (KE) contribute to the understanding of the nervous system, cognitive processes and social behavior? This is the scope of computational neuroscience and cognition, which uses the computational paradigm to model and improve our understanding of natural science.

II: From Natural Sciences to Computation, AI and KE. How can computation, AI and KE find inspiration in the behavior and internal functioning of physical, biological and social systems to conceive, develop and build up new concepts, materials, mechanisms and algorithms of potential value in real-world applications? This is the scope of the new bionics, known as bioinspired engineering and computation, as well as of natural computing.

To address the two questions exposed in the scope of IWINAC 2011, we will make use of a wide and comprehensive view of the computational paradigm that first considers three levels of description for each calculus (physical mechanisms, symbols and knowledge) and then distinguishes between two domains of description (the level "own" domain and the domain of the external observer).

This wider view of the computational paradigm allows us more elbow room to accommodate the results of the interplay between nature and computation. The IWINAC forum thus becomes a methodological approximation (set of intentions, questions, experiments, models, algorithms, mechanisms, explanation procedures, and engineering and computational methods) to the natural and artificial perspectives of the mind embodiments problem, both in humans and in artifacts. This is the philosophy of the IWINAC meetings, the "interplay" movement between the natural and artificial, facing this same problem every two years. We want to know how to model biological processes that are associated with measurable physical magnitudes and, consequently, we also want to design and build robots that imitate the corresponding behaviors based on that knowledge. This synergistic approach will permit us not only to build new computational systems based on the natural measurable phenomena, but also to understand many of the observable behaviors inherent to natural systems.

The difficulty of building bridges over natural and artificial computation was one of the main motivations for the organization of IWINAC 2011. In this edition, the conference was simultaneously coorganized with the Joint Workshop and Summer School: Astrostatistics and Data Mining in Large Astronomical Databases 2011, that aims to apply AI techniques to astronomical data. The IWINAC 2011 proceedings volumes include the 108 works selected by the Scientific Committee after a refereeing process. The first volume, entitled *Foundations on Natural and Artificial Computation*, includes all the contributions mainly related to the methodological, conceptual, formal, and experimental developments in the fields of neurophysiology and cognitive science. The second volume entitled *New Challenges on Bioinspired Applications* contains the papers related to bioinspired programming strategies and all the contributions related to the computational solutions to engineering problems in different application domains, especially health applications, including the CYTED "Artificial and Natural Computation for Health" (CANS) research network papers.

An event like IWINAC 2011 cannot be organized without the collaboration of a group of institutions and people, whom we would like to thank now, starting with *UNED* and *Universidad Politécnica de Cartagena*. The collaboration of the *UNED associated center* was crucial, as was the efficient work of the Local Committee, Francisco Javier Neris Paz and Juan Antonio González Arnaez, with the close collaboration of the *Instituto de Astrofísica de Canarias*, and the essential support of Rafael Rebolo and Juan Carlos Pérez. In addition to our universities, we received financial support from the Spanish *Ministerio de Ciencia e Innovación, CYTED, Red Nacional en Computación Natural y Artificial* and *APLIQUEM S.L.*

We want to express our gratefulness to our invited speakers, Changjiu Zhou from Singapore Polytechnic, Paul Cull, Oregon State University, Rüdiger Dillmann from Karlsruhe Institute of Technology (KIT) and Jon Hall, Open University, for accepting our invitation and for their magnificent plenary talks.

We would also like to thank the authors for their interest in our call and the effort in preparing the papers, condition *sine qua non* for these proceedings, and to all the Scientific and Organizing Committees, in particular, the members of these committees that have acted as effective and efficient referees and as promoters and managers of pre-organized sessions on autonomous and relevant topics under the IWINAC global scope.

Our sincere gratitude also goes to Springer and to Alfred Hofmann and his collaborators, Anna Kramer and Leonie Kunz, for the continuous receptivity, help, and collaboration in all our joint editorial ventures on the interplay between neuroscience and computation.

Finally, we want to express our special thanks to *ESOC S.L.*, our technical secretariat, and to Victoria Ramos, for making this meeting possible, arranging all the details that comprise the organization of this kind of event.

All the authors of papers in this issue, as well as the IWINAC Program and Organizing Committees, would like to pay tribute to the memory of Professor Mira, both as a great scientist and as a good friend. We still greatly miss him.

June 2011

José Manuel Ferrández Vicente
José Ramón Álvarez Sánchez
Félix de la Paz López
Fco. Javier Toledo Moreo

Organization

General Chairman

José Manuel Ferrández Vicente

Organizing Committee

José Ramón Álvarez Sánchez
Félix de la Paz López
Fco. Javier Toledo Moreo

Local Organizing Committee

Francisco Javier Neris Paz
Juan Antonio González Arnaez

Invited Speakers

Paul Cull, USA
Rüdiger Dillmann, Germany
Jon Hall, UK
Changjiu Zhou, Singapore

Field Editors

Diego Andina, Spain
José M. Azorín, Spain
Mª Consuelo Bastida Jumilla, Spain
Francisco Bellas, Spain
Carlos Cotta Porras, Spain
Verónica Dahl, Canada
Richard Duro, Spain
Eduardo Fernández, Spain
Antonio Fernández Caballero, Spain
Antonio J. Fernández Leiva, Spain
José Manuel Ferrández, Spain
Vicente Garcerán Hernández, Spain
Pedro Gómez Vilda, Spain
Juan Manuel Górriz Sáez, Spain
M. Dolores Jiménez, Spain
Javier de Lope, Spain

Dario Maravall, Spain
Rafael Martínez Tomás, Spain
Félix de la Paz, Spain
Mariano Rincón Zamorano, Spain
Javier Ropero Peláez, Brazil
Daniel Ruiz Fernández, Spain
Andreas Schierwagen, Germany
Antonio Soriano, Spain
M. Jesús Taboada, Spain

Scientific Committee (Referees)

Andy Adamatzky, UK
Michael Affenzeller, Austria
Abraham Ajith, Norway
Igor Aleksander, UK
Amparo Alonso Betanzos, Spain
Jose Ramon Alvarez Sanchez, Spain
Shun ichi Amari, Japan
Diego Andina, Spain
Davide Anguita, Italy
Margarita Bachiller Mayoral, Spain
Antonio Bahamonde, Spain
Dana Ballard, USA
Emilia I. Barakova, The Netherlands
Alvaro Barreiro, Spain
Senen Barro Ameneiro, Spain
Francisco Bellas, Spain
Guido Bologna, Switzerland
Juan Botia, Spain
François Bremond, France
Giorgio Cannata, Italy
Enrique J. Carmona Suarez, Spain
Joaquin Cerda Boluda, Spain
Enric Cervera Mateu, Spain
Antonio Chella, Italy
Santi Chillemi, Italy
Eris Chinellato, Spain
Emilio S. Corchado, Spain
Carlos Cotta, Spain
Erzsebet Csuhaj Varju, Hungary
Jose Manuel Cuadra Troncoso, Spain
Veronica Dahl, Canada
Felix de la Paz Lopez, Spain
Javier de Lope, Spain
Erik De Schutter, Belgium

Angel P. del Pobil, Spain
Ana E. Delgado Garcia, Spain
Gines Domenech, Spain
Jose Dorronsoro, Spain
Gerard Dreyfus, France
Richard Duro, Spain
Reinhard Eckhorn, Germany
Patrizia Fattori, Italy
Juan Pedro Febles Rodriguez, Cuba
Paulo Felix Lamas, Spain
Eduardo Fernandez, Spain
Antonio Fernandez Caballero, Spain
Manuel Fernandez Delgado, Spain
Miguel A. Fernandez Graciani, Spain
Antonio J. Fernandez Leiva, Spain
Abel Fernandez Laborda, Spain
Jose Manuel Ferrandez, Spain
Kunihiko Fukushima, Japan
Cristina Gamallo Solorzano, Spain
Jose A. Gamez, Spain
Vicente Garceran Hernandez, Spain
Jesus Garcia Herrero, Spain
Juan Antonio Garcia Madruga, Spain
Francisco J. Garrigos Guerrero, Spain
Tamas D. Gedeon, Australia
Charlotte Gerritsen, The Netherlands
Marian Gheorghe, UK
Pedro Gomez Vilda, Spain
Juan M Gorriz, Spain
Manuel Graña Romay, Spain
Francisco Guil Reyes, Spain
John Hallam, Denmark
Juan Carlos Herrero, Spain
Cesar Hervas Martinez, Spain
Tom Heskes, The Netherlands
Roberto Iglesias, Spain
Fernando Jimenez Barrionuevo, Spain
M. Dolores Jimenez Lopez, Spain
Kok Joost N., The Netherlands
Jose M. Juarez, Spain
Kostadin Koroutchev, Spain
Elka Korutcheva, Spain
Yasuo Kuniyoshi, Japan
Petr Lansky, Czech Republic
Markus Lappe, Germany

Maria Longobardi, Italy
Maria Teresa Lopez Bonal, Spain
Ramon Lopez de Mantaras, Spain
Pablo Lopez Mozas, Spain
Tino Lourens, The Netherlands
Max Lungarella, Japan
Manuel Luque Gallego, Spain
Francisco Macia Perez, Spain
George Maistros, UK
Saturnino Maldonado, Spain
Vincenzo Manca, Italy
Daniel Mange, Switzerland
Riccardo Manzotti, Italy
Dario Maravall, Spain
Roque Marin, Spain
Jose Javier Martinez Alvarez, Spain
Rafael Martinez Tomas, Spain
Jesus Medina Moreno, Spain
Jose del R. Millan, Switzerland
Victor Mitrana, Spain
Jose Manuel Molina Lopez, Spain
Javier Monserrat Puchades, Spain
Juan Morales Sanchez, Spain
Federico Moran, Spain
Roberto Moreno Diaz, Spain
Arminda Moreno Diaz, Spain
Ana Belen Moreno Diaz, Spain
Isabel Navarrete Sanchez, Spain
Nadia Nedjah, Brazil
Taishin Y. Nishida, Japan
Richard A. Normann, USA
Manuel Ojeda Aciego, Spain
Lucas Paletta, Austria
Jose T. Palma Mendez, Spain
Juan Pantrigo, Spain
Alvaro Pascual Leone, USA
Miguel Angel Patricio Guisado, Spain
Gheorghe Paun, Spain
Juan Pazos Sierra, Spain
Mario J. Perez Jimenez, Spain
Jose Manuel Perez Lorenzo, Spain
Franz Pichler, Austria
Jose M. Puerta, Spain
Carlos Puntonet, Spain
Alexis Quesada Arencibia, Spain

Andonie Razvan, USA
Luigi M. Ricciardi, Italy
Mariano Rincon Zamorano, Spain
Victoria Rodellar, Spain
Jesus Rodriguez Presedo, Spain
Jose Carlos Rodriguez Rodriguez, Spain
Camino Rodriguez Vela, Spain
Javier Ropero Pelaez, Brazil
Daniel Ruiz, Spain
Ramon Ruiz Merino, Spain
Pedro Salcedo Lagos, Chile
Juan Vicente Sanchez Andres, Spain
Angel Sanchez Calle, Spain
Eduardo Sanchez Vila, Spain
Jose Luis Sancho Gomez, Spain
Gabriella Sanniti di Baja, Italy
Jose Santos Reyes, Spain
Ricardo Sanz, Spain
Shunsuke Sato, Japan
Andreas Schierwagen, Germany
Guido Sciavicco, Spain
Radu Serban, The Netherlands
Igor A. Shevelev, Russia
Juan A. Sigüenza, Spain
Jordi Solé i Casals, Spain
Antonio Soriano Paya, Spain
Maria Jesus Taboada, Spain
Settimo Termini, Italy
Javier Toledo Moreo, Spain
Rafael Toledo Moreo, Spain
Jan Treur, The Netherlands
Enric Trillas Ruiz, Spain
Ramiro Varela Arias, Spain
Marley Vellasco, Brazil
Lipo Wang, Singapore
Stefan Wermter, UK
Hujun Yin, UK
Changjiu Zhou, Singapore

Table of Contents – Part I

Table of Contents – Part II

Complex Neuro-Cognitive Systems

Andreas Schierwagen

Institute for Computer Science, Intelligent Systems Department,
University of Leipzig, Leipzig, Germany
schierwa@informatik.uni-leipzig.de
http://www.informatik.uni-leipzig.de/~schierwa

Abstract. Cognitive functions such as a perception, thinking and acting are based on the working of the brain, one of the most complex systems we know. The traditional scientific methodology, however, has proved to be not sufficient to understand the relation between brain and cognition. The aim of this paper is to review an alternative methodology – nonlinear dynamical analysis – and to demonstrate its benefit for cognitive neuroscience in cases when the usual reductionist method fails.

1 Introduction

Cognitive science aims at both understanding natural cognition (as in humans or animals) and creating artificial systems resembling the natural original. There are basically two established ways of doing research in cognitive science (including cognitive and computational neuroscience), either top-down or bottom-up. Both approaches have their pro's and con's, and none is practised in the pure form. For example, the main research strategy in computational neuroscience - reverse engineering - is basically bottom-up. It is informed, however, by top-down considerations about the goal of the computation performed by the neural system under study. It is expected that united efforts of this kind will succeed in providing a theory of cognition that is ultimately grounded in brain processes.

However, objections have been also raised saying that the established methods do not meet the complexity of the object of study, and that the research methodolgy must be complemented accordingly. In this paper, I'll first describe briefly the two traditional ways of doing cognitive (neuro-)science that are commonly thought to exhaust the possibilities. Then a third way of analysis - nonlinear dynamical analysis – is described that relies on results of complexity research, i.e. chaos theory and nonlinear modeling.

2 Top-Down and Bottom-Up Approaches

The top-down approach consists of (1) specifying a cognitive function by focusing on the characterization of the abstract principles that underlie that function. Ideally, it proposes (2) possible neural algorithms that might subserve this cognitive function, and finally (3) maps these algorithms onto brain circuits. In many

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 1–9, 2011.

cases, however, the identification steps (2) and (3)have proved to be very difficult or unfeasible. The bottom-up approach consists of describing the structural and functional properties of given brain circuits, and then bringing this function into congruence with the cognitive function under study.

Previously [1,2], I have shown that both approaches set up the method of *reverse engineering*. This method combines analysis with synthesis in the following way. Analysis is carried out top-down by specifying first a certain cognitive function which is assumed to be computed through the cortex or some cortical subsystem. Then a decompositional analysis is performed, i.e. the cortical system is both functionally (computationally) and structurally decomposed, and the interactions between components are determined. Following the localisation concept, the functional components (computational units) are assigned to the anatomical components.

Synthesis first requires modelling, i.e building a structurally adequate, functional model of the computational unit. Based on knowledge of localised components and their interactions, a structurally adequate network model of the cortical system is build composed of the computational unit models. Simulations of the network model should eventually prove that the specific cognitive function under study is generated this way.

Recent efforts to build *artificial brains*[1] employ both approaches [6,7]. Large-scale brain simulations attempt to model in a realistic fashion the details of the brain organisation, i.e. its structure and function. The Blue Brain Project is one prominent example of this bottom-up modelling strategy; its explicit goal is to reverse engineer the brain. On the other hand, biologically inspired cognitive architectures (BICAs) rely on the top-down approach. They attempt to achieve the brain's cognitive functionality by emulating its high-level performance without capturing the neural details.

In the survey [6,7] it was concluded that the two approaches display very different strengths. While bottom-up brain simulations are confined to syntactic aspects like how collections of neurons synchronize their electrical discharges, they do not tell anything about semantics, i.e. how brain processes enable cognitive agents to achieve goals, select actions or process information. In contrast, BICAs propose how brains may realise cognitive functions, but as yet they demonstrate rather simplistic behaviour compared to real brains. The authors conjecture that the deficiencies may be due to the fact that "BICAs lack the chaotic, complex generativity that comes from neural nonlinear dynamics - i.e. they have the sensible and brain-like higher-level structures, but lack the lower-level complexity and emergence that one sees in large-scale brain simulations." [7, p. 48]. Bringing large-scale brain simulations and BICAs together, they suggest, will accomplish progress toward the goals of cognitive science - understanding the brain and creating artificial cognitive systems.

[1] "Artificial brain" is a term used to describe research that aims to develop software and hardware with cognitive abilities similar to humans or other mammals. Prominent examples are SyNAPSE (Systems of Neuromorphic Adaptive Plastic Scalable Electronics)[3], the Blue Brain Project[4] and the China Brain Project [5].

3 Complex Systems

The suggestion to integrate bottom-up, large-scale brain simulations and top-down theories such as BICAs to progress in neuro-cognition research has been made from time to time. However, the predicted success has not appeared what is apparently due to the fact that both approaches have restrictions which cannot be overcome even by integration. Actually, both analysis methods are applicable only to a limited class of systems, the (near-)decomposable systems, as shown elsewhere [1,2]. There I have argued that the subjects of study of cognitive and computational neuroscience – cognitive systems that realise functions localised in neural circuits of the brain – are not members of this class. They are instances of *complex systems* which resist the usual reductionist analyses.

The study of complex systems originated during the last three decades or so from the interplay of disciplines such as physics, mathematics, biology, economy, engineering, and computer science. There is still no generally accepted definition of complexity, despite a multitude of proposed approaches (e.g. [8,9,10]).

Important is the following distinction: we must differentiate between systems that are complex and those which are merely complicated [11, p.511]:

> *A complicated system is composed of a large number of interacting components. Importantly, the properties of such a system can be accurately predicted from a knowledge of the properties of each of its components and a complete enumeration of their interactions. In other words, a complicated system is exactly the sum of its parts. Complex, on the other hand, is a term reserved for systems that display properties that are not predictable from a complete description of their components, and that are generally considered to be qualitatively different from the sum of their parts.*
>
> Editorial. Nature Biotechnology, 1999

From complexity theory we know that complex phenomena can be produced from the interaction of rather simple components. Well-understood examples are artificial neural networks and cellular automata. These are compositionally complex systems, and it is indeed feasible to predict their behaviour from the knowledge of the properties of the components and their interactions. This is completely different in the case of complex systems whose behaviour emerges in an unpredictable way. The question then arises: how should complex systems be studied, and which methods of investigation are available? In the following, I will consider the method of *nonlinear dynamical analysis*. Other methods include relational modelling [12,13] and quantum theories [14].

4 Nonlinear Dynamical Analysis

4.1 Terminology

To apply a nonlinear dynamical analysis to a complex system means to specify its temporal evolution. The *state space* (or *phase space*) of the system consists of the set of all its possible states, s, determined by the values of all the variables that describe the system at a particular moment in time. If s is described by m variables, it can be represented by a point in m-dimensional space, $s \in R^m$. The system dynamics is given by the set of equations or rules that control the system evolution over time. In many cases the dynamics consists of a system of m coupled differential equations, one for each system variable. Most natural systems are continuous systems and therefore s is a function of time, $s(t)$. As the system evolves in time its states trace a *trajectory* in the state space.

An attractor in state space may be defined as a state (*point* attractor) or set of states, toward which the system settles (relaxes) over time. Besides point attractors, three more attractor types can occur: (i) limit cycles; (ii) torus attractors; (iii) chaotic attractors. A *limit cycle* is a closed trajectory (an *orbit*) in state space that the system performs cyclically; when a system evolves towards a periodic attractor, it will oscillate endless through the same sequence of states (unless perturbed). A torus attractor has a 'donut like' shape, and corresponds to quasi periodic dynamics. A *chaotic* attractor is a non-repeating orbit in state space, i.e. the system dynamics, although deterministic, will never repeat the same state; it is called *deterministic chaos*.

Several measures are used to characterize the properties of attractors, and thus of the corresponding dynamics more exactly. *Correlation dimension* is a measure of the complexity of the deterministic dynamics. A point attractor has dimension zero, a limit cycle dimension one, a torus has an integer dimension corresponding to the number of superimposed periodic oscillations, and a chaotic attractor has a fractal dimension. *Lyapunov exponents* indicate the exponential divergence (positive exponents) or convergence (negative exponents) of nearby trajectories on the attractor, thus giving information about the systems dependence on initial conditions. A positive Lyapunov exponent is a strong indicator of chaos.

4.2 An Example

Complex behaviour (*dynamical complexity*) can arise even in simple systems with low compositional complexity. The damped, periodically driven non-linear

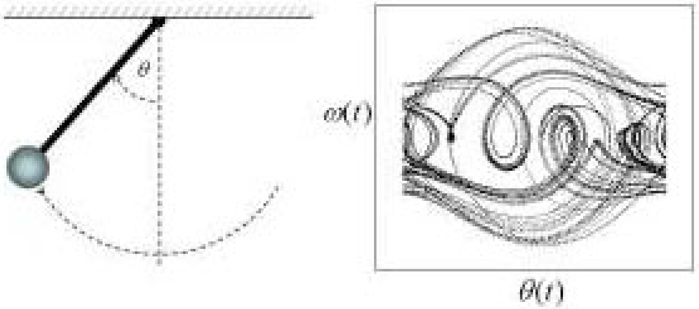


Fig. 1. The damped, periodically driven non-linear pendulum. Displayed is the chaotic attractor in the (θ, ω)–phase space that occurs for large values of the driving force g.

pendulum is suited for illustrating the principles behind complex dynamics and chaotic attractors (Fig. 2). The dimensionless motion equation of the pendulum is:

$$\frac{d^2\theta}{dt^2} + \frac{1}{q}\frac{d\theta}{dt} + \sin\theta = g\cos(\omega_D t) \tag{1}$$

with θ the angular position in radians, q the damping parameter, g the amplitude of the driving force, and ω_D the frequency of that force. For small angle θ the equation can be integrated, i.e. the pendulum either undergoes regular oscillations or, without a driving term, eventually stop swinging. For larger angles θ, however, this approximation is invalid and hence the equation can no longer be solved analytically.

The dynamical variables of the system, i.e. the angular position θ and velocity $\omega = d\theta/dt$, are the coordinates defining the system's *phase space*. In the two–dimensional case, the variables can be plotted to display a *phase portrait* of the dynamical behaviour.

By varying the parameters q, g and ω_D of the motion equation (1) and then plotting the resulting phase portrait a wide range of behaviour can be observed. In the case of a non-zero damping parameter and no driving force to replace the energy loss, the pendulum is a *dissipative* system, i.e. it comes to the resting point (0,0), a fixpoint attractor. Using a non-zero driving force g, the attractor is no longer a single point at (0,0) but now a closed orbit, that is, the pendulum undergoes regular motion. A bifurcation has occurred, changing the fixpoint into a limit cycle attractor. Increasing the driving force g further, a sequence of period doublings occurs which continues as g is increased until a point is reached where the motion of the pendulum ceases to be regular and becomes chaotic, i.e. a chaotic attractor occurred.

This example nicely shows that even in a compositionally very simple system the dynamics can be chaotic. In this case, the analysis could be made because the dynamical system model, i.e. the equation of motion (1), is known.

4.3 Nonlinear Time Series Analysis

We have discussed complex systems with nonlinear dynamics so far using the bottom-up, deductive approach: given the system equations, the behaviour of the system can be predicted. However, the situation in cognitive neuroscience is totally different: instead of the system equations, a set of observations is given, say in the form of an EEG record. The problem then is to find a way to get from the observations of a system with unknown properties to an understanding of the dynamics of the underlying system. This can be achieved by *nonlinear time series analysis*, a systemic approach starting with the output of the system, and working towards the state space, attractors and their properties.

Nonlinear time series analysis proceeds as follows: (i) reconstruction of the systems dynamics in state space; (ii) characterization of the reconstructed attractor; (iii) checking the validity of the procedure [17].

State space reconstruction. In order to reconstruct the state space using a time series, it is resolved into coordinate values of a d-dimensional embedding space by an embedding method. Let the state space be characterized by the set of variables, $\{x_0(t), x_1(t), ..., x_{d-1}(t)\}$. Most frequently used is *time delay embedding*. Assume that only time series $x_0(t)$ is available[2]. Then time-delayed values of this series are used,

$$\{x_0(t), x_0(t+\tau), ..., x_0[t+(d-1)\tau]\} .$$

This set is topologically equivalent to the original set of system variables, see [18]. These variables are obtained by shifting the original time series by a fixed time lag $\tau = m\Delta t$ where m is an integer and Δt is the interval between successive samplings. A most important problem of state space reconstruction is the determination of the delay time τ and embedding dimension d for which several methods are available. This result is known as Taken's famous 'Embedding Theorem' which says: valuable information about the dynamics of the system can be obtained, even if direct access to all the systems variables is impossible, as it is common in cognitive neuroscience!

Characterization of the reconstructed attractor. After reconstruction of the attractor by embedding the next step is to characterize it. A common way to do this is to visualize it with a phase portrait. A phase portrait is a two– or three–dimensional plot of the reconstructed state space and the attractor. The graph shown in Fig. 1 is an example of a two–dimensional phase portrait. Other methods to display the reconstructed trajectories are Poincaré sections and recurrence plots [19].

Following embedding and visualization of the reconstructed attractor the next step is to attempt to characterize it in a quantitative way. The classic measures applied are correlation dimension, Lyapunov exponents and entropy mentioned in Section 4, and new measures introduced frequently in the literature.

Checking the validity of the procedure. The interpretation of nonlinear measures is known to present problems sometimes since noisy time series can give rise to the unwarranted impression of low–dimensional dynamics and chaos. Therefore, the nonlinearity of the time series should be tested. It is customary to do this by *surrogate data test*. The null hypothesis of the test is that the original time series is generated from a linear stochastic process (possibly undergoing a nonlinear static transform). Demonstration of nonlinearity is important since only nonlinear dynamical systems can have attractors other than a trivial fixpoint attractor. Chaos in particular can only occur in nonlinear dynamical systems.

5 Explanation by Nonlinear Dynamic Analysis

In the preceding sections the methodology of nonlinear dynamic analysis was outlined roughly. We are now prepared to demonstrate how it is used in cognitive

[2] x_0 may represent the electrical potential recorded by the EEG.

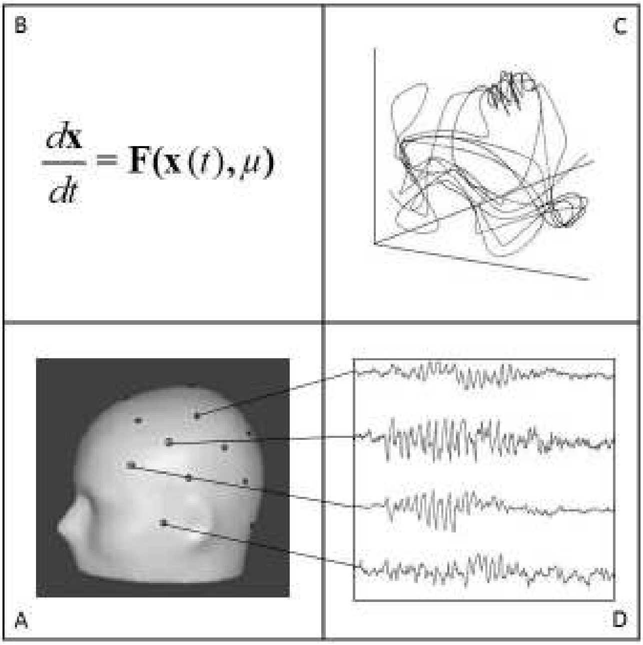


Fig. 2. Reductionist, deductive explanation is illustrated by following the path through cells $A \to B \to C \to D$. This approach fails in non-decomposable systems with complex dynamics since no system equations can be established. Instead, observable behaviour of the real system is explained by nonlinear dynamic analysis, i.e. move $C \to D$ is to be made (inspired by [16]).

neuroscience. Let us consider the situation presented in Fig. 2. In cell A the physical system under study is displayed (a human's head with EEG recording sites). Cell B shows a formal model (a system of nonlinear differential equations). Establishing the model requires that all the variables determining the system and their dynamic connections are exactly known. In cell C the phase portrait in the system state space is schematically illustrated. It allows to describe the possible system behaviour in terms of trajectories, attractors, bifurcations etc. Cell D presents the observed system behaviour which is often measured in cognitive science in the form of time series (in this case the EEG activity recorded via several channels).

The usual kind of reductionist, deductive explanation can be illustrated by following the path through cells $A \to B \to C \to D$ in Fig. 2. The behaviour of the physical system as measured at the emergent level is reduced via the formal model to the lower level of the physical substrate. The transitions between the fields are all non-trivial. Move $A \to B$ requires to determine the relevant system variables and to study their dynamics which is rather unfeasible in the case of neuro-cognitive systems. Move $B \to C$ means nonlinear dynamical analysis of system equations which may pose serious difficulties, depending on system characteristics. Finally, by moving $C \rightarrow D$ the formal constructs of nonlinear system theory are to be mapped on real, observable phenomena in the neurocognitive system. This move again is very challenging, and cannot be formalised [12].

Emergent phenomena of the real, dynamically complex system must be explained using nonlinear dynamic analysis since traditional analyses fail. That is, in the scheme of Fig. 2 only move $C \rightarrow D$ is (and can be) made. A meanwhile famous example is the work by Babloyantz and Destexhe [15] who demonstrated that the epileptic electric brain activity measured by the EEG forms a chaotic attractor. Here the Lyapunov exponents of the attractors and the embedding dimension of the phase space was calculated from the time series of the EEG. This was done without having available a formal model. From the mere fact that a chaotic attractor with certain mathematical properties is present, non-trivial conclusions (e.g., on stability, dimensionality of embedding space etc.) has been drawn.

Systemic explanations of this kind have explanatory power because dynamically complex systems own certain universal properties. If a whole class of dynamic systems is characterised by a certain qualitative property, it is unnecessary to know the exact form of the special dynamic model which explains the particular emergent phenomenon. It is only necessary to assure that the system under study belongs to the corresponding class of dynamic systems in which certain qualitative phenomena are universal. This kind of the explanation works because classes of dynamically complex systems own qualitative properties (attractor types, bifurcations, pattern generation, chaos).

6 Conclusions

The progress made with analyses of compositionally complex (complicated) systems such as artificial neural networks, cellular automata etc. has led many to believe that this can be achieved with dynamically complex systems, too. This means that the reverse engineering method has to be applied to the observed dynamically complex phenomenon to work out which mechanism explains it best. A mechanism is known if the participating system components and their interactions are known, i.e. the term 'mechanism' is nearly synonymous with 'decomposability'. If we remember that dynamically complex systems are not decomposable, it follows that no mechanism can be revealed, and the reductionist analysis fails!

In this vein, proponents of the synthetic bottom-up simulation approach often argue that a systemic, non–reductionist approach such as nonlinear dynamical analysis merely provides provisional solutions which become only then full explanations when they are complemented by a formal model and by the physical basis of the empirical system. However, this view again ignores the specificity of the dynamically complex systems which do not allow reductionist, mechanistic explanation. Thus, nonlinear dynamic analyses provide full-value explanations which are highly appropriate to the situation in cognitive neuroscience.

References

1. Schierwagen, A.: Brain complexity: analysis, models and limits of understanding. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5601, pp. 195–204. Springer, Heidelberg (2009)
2. Schierwagen, A.: On reverse engineering in the cognitive and brain sciences. Natural Comput (2011) (in press)
3. Systems of Neuromorphic Adaptive Plastic Scalable Electronics (SyNAPSE). DARPA / IBM (2008)
4. Markram, H.: The Blue Brain Project. Nature Rev. Neurosci. 7, 153–160 (2006)
5. de Garis, H., et al.: The China-Brain Project: Building China's Artificial Brain Using an Evolved Neural Net Module Approach. In: Wang, P., Goertzel, B., Franklin, S. (eds.) Proceedings First AGI Conference, pp. 107–121. IOS Press, Amsterdam (2008)
6. de Garis, H., Shuoa, C., Goertzel, B., Ruiting, L.: A world survey of artificial brain projects, Part I: Large-scale brain simulations. Neurocomput. 74, 3–29 (2010)
7. Goertzel, B., Ruiting, L., Arel, I., de Garis, H., Chen, S.: World survey of artificial brains, Part II: Biologically inspired cognitive architectures. Neurocomput. 74, 30–49 (2010)
8. Edmonds, B.: Syntactic Measures of Complexity. PhD thesis, University of Manchester (1999)
9. Chu, D., Strand, R., Fjelland, R.: Theories of complexity. Complexity 8, 19–30 (2003)
10. Gershenson, C.: Complexity. arXiv:1003.5947v1
11. Editorial. Complicated is not complex. Nature Biotechnology 17, 511 (1999)
12. Rosen, R.: Life Itself: A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life. Columbia University Press, New York (1991)
13. Rosen, R.: Essays on Life Itself. Columbia University Press, New York (2000)
14. Kitto, K.: High End Complexity. Intern. J. Gen. Syst. 37, 689–714 (2008)
15. Babloyantz, A., Destexhe, A.: Low-dimensional chaos in an instance of epilepsy. Proc. Natl. Acad. Sci. USA 83, 3513–3517 (1986)
16. Jaeger, H.: Dynamische Systeme in der Kognitionswissenschaft. Kognitionswissenschaft 5, 151–174 (1996)
17. Stam, C.J.: Nonlinear dynamical analysis of EEG and MEG: Review of an emerging field. Clin. Neurophysiol. 116, 2266–2301 (2005)
18. Takens, F.: Detecting strange attractors in turbulence. Lecture Notes Math. 898, 366–381 (1981)
19. Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis. Cambridge University Press, Cambridge (1997)

A Markov Model of Conditional Associative Learning in a Cognitive Behavioural Scenario

Stefan Glüge[1], Oussama H. Hamid[2], Jochen Braun[2], and Andreas Wendemuth[1]

[1] Faculty of Electrical Engineering and Information Technology,
Otto von Guericke University Magdeburg,
Universitätsplatz 2, 39106 Magdeburg, Germany
stefan.gluege@ovgu.de

[2] Cognitive Biology, Institute of Biology,
Otto von Guericke University Magdeburg, Leipziger Strasse 44,
39120 Magdeburg, Germany
jochen.braun@ovgu.de

Abstract. In conditional learning, one investigates the computational principles by which the human brain solves challenging recognition problems. The role of temporal context in the learning of arbitrary visuomotor associations has so far been studied mostly in primates. We model the explicit learning task where a sequence of visual objects is presented to human subjects. The computational modelling of the algorithms that appear to underlie human performance shall capture the effects of confusion in ordered and random presentation of objects. We present a Markov model where the learning history of a subject on a certain object is represented by the states of the model. The analysis of the resulting Markov chain makes it possible to judge the influence of two model parameters without the simulation of a specific learning scenario. As the model is able to reproduce the learning behaviour of human subjects it might be useful in the development of future experiments.

1 Introduction

In biology, conditional associative learning is often studied in the context of arbitrary sensorimotor mappings [9,2]. Typically, the experimental design takes a set of visual stimuli from the same category and maps them randomly onto a set of motor responses. Subjects learn by trial and error which response produces the reward in the case of each stimulus. As all stimuli are potentially associated with reward, the subject cannot simply learn stimulus-reward associations. Instead, subjects must link each stimulus to the specific response that ensures the reward in each case.

The complexity of this task implies that multiple brain areas may be involved. In fact, studies with behaving non-human primates reveal an extensive network of brain regions underlying conditional associative learning [10,1,8].

To study the effect of temporal order on conditional associative learning, we modified Miyashita's classical experiments with non-human primates [6,7]. Our

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 10–19, 2011.

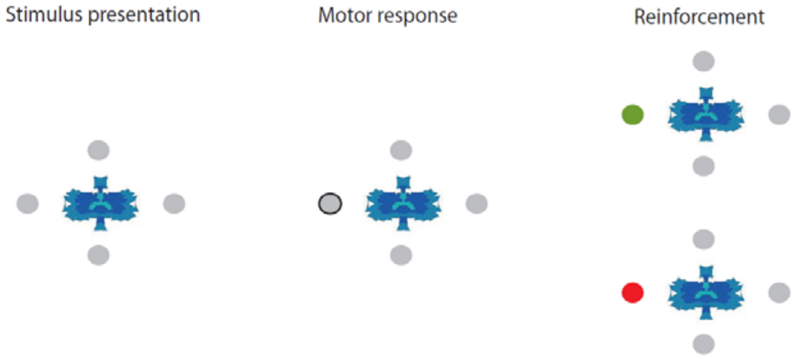


Fig. 1. Experimental design (schematic). Each trial comprises three phases: stimulus presentation, motor response, and reinforcement. Firstly, a fractal object appears, surrounded by four response options (grey discs). Secondly, the observer reacts by pressing the key that corresponds to one response option (outlined disk). Thirdly, a colour change of the chosen option provides reinforcement (green if correct, red if incorrect). [5]

human observers viewed highly distinguishable, fractal objects and learned to select one of four possible motor responses for each object. Some objects were consistently preceded by specific other objects, while other objects lacked such temporal context.

Observers were instructed to learn to respond 'correctly' to each fractal object. It was explained that, for each fractal object, one of the four possible responses was 'correct', while the other three responses were 'incorrect'. Observers were told that they had to become familiar with and learn to recognise each fractal object and that they had to learn the 'correct' response of each object by trial and error (Fig. 1). They were further told that there was no pattern or system that would enable them to predict which response a particular fractal object required. No mention of or reference to the sequence of trials and fractal objects was made.

Sequences contained eight objects and were either maximally deterministic or maximally random. In the deterministic sequence, the eight objects always appeared in the same order, so that preceding objects were just as predictive about the correct response in the current trial as the current object. In the random sequence, each object followed every other object with equal probability. Thus, preceding objects provided almost no information about the correct response in the current trial.

Observers quickly understood the existence and nature of the two types of sequences (even though the instructions had been silent on this point). These findings show that temporal context significantly accelerates conditional associative learning. Further details about this and four additional experiments have been reported in [5]. Further, a reinforcement learning model was introduced in [5] and a neural network model in [3]. Both models aim to investigate the implicit learning of the temporal context. In contrast to these approaches, we developed a Markov model for the explicit task of associative learning (object association with the correct button). It ignores the temporal context between objects and gives the chance to an analysis of the associative learning process.

The remaining parts of the paper are organised as follows. Section 2 introduces the behavioural Markov model along the line of the underlying experiment. In Sect. 3 we derive the significance of two parameters by an analysis of the Markov chain, representing the associative learning task. Finally, in Sect. 4, we present the results of a simulation with the model and discuss the relevance of the findings.

2 Behavioural Markov Model

The Markov property of a random process says, that the next state in the process depends *only* on the current state. A Markov model is a particular type of Markov process in which the process consists of a finite number of states.

We developed a Markov model for the given conditional learning task. The effects of distinguishing objects, memorising previous actions, and memorising success for an object are represented by separate parameters. The history of the learning process (subject's actions) is accumulated in the current state. The basic assumption is, that subjects follow a rational policy in choosing the action on a certain object. This is: (i) once the successful button was found, this choice is repeated in later trials and (ii) the choices which so far have not been successful are not taken again, and the remaining options are chosen with equal probability.

Following these policy, we define the Markov model for learning the motor response for an object by trial and error. The model consists of 16 states ($S = \{s_1, s_2, \dots, s_{16}\}$). Each represents one possible situation a subject could face during learning. At the beginning the subject has no clue about any of the four possible buttons (s_1). For each possible decision we define four states ($s_2, \dots, s_5$). After the first choice the subject has three possibilities left, which lead to the next states and so on. Table 1 shows all possible situations/states whereas 'x' implies 'chosen and memorised to be wrong' and '.' implies 'not chosen yet'. The last state s_{16} represents success. In this state the subject found the correct button and memorised it.

Knowing the state space of the learning process, we have to define the way through this space during learning. This is described by transition probabilities, summarised in the state transition matrix $\mathbf{T}$. The general structure of the matrix

Table 1. Possible states during associative learning of an object by trial and error

state s	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
left	.	x	.	.	.	x	x	x	.	.	.	x	x	x	.	success
right	.	.	x	.	.	x	.	.	x	x	.	x	x	.	x	success
up	.	.	.	x	.	.	x	.	x	.	x	x	.	x	x	success
down	.	.	.	.	x	.	.	x	.	x	x	.	x	x	x	success

comprises the policy a subject follows. At this point, we introduce a parameter which allows to make a mistake in recognition of success (third phase – reinforcement, cf. Fig. 1). If the system indicates that the chosen button was correct, the subject might memorise this event. On the other hand, it may simply forget this event in the following. This uncertainty is represented by the *success recall probability* p_{suc} indicating the chance a subject memorises a success. Hence, p_{suc} is the probability to get into the success state (s_{16}) if the correct button was pressed. The ideal case denotes $p_{\mathrm{suc}} = 1$. A lower p_{suc} lowers the ability of the model to recall a success. With this parameter we find the transition matrix to take the form

$$\mathbf{T}_{\mathrm{left}} = \begin{pmatrix}
0 & \frac{1-p_{\mathrm{suc}}}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p_{\mathrm{suc}}}{4} \\
0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1-p_{\mathrm{suc}}}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{p_{\mathrm{suc}}}{3} \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p_{\mathrm{suc}}}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{p_{\mathrm{suc}}}{3} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p_{\mathrm{suc}}}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{p_{\mathrm{suc}}}{3} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p_{\mathrm{suc}}}{2} & 0 & 0 & \frac{1}{2} & \frac{p_{\mathrm{suc}}}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p_{\mathrm{suc}}}{2} & 0 & \frac{1}{2} & \frac{p_{\mathrm{suc}}}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-p_{\mathrm{suc}}}{2} & \frac{1}{2} & \frac{p_{\mathrm{suc}}}{2} \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} . \tag{1}$$

Equation (1) shows the transition matrix in case that the button ‘left’ is correct for the concerning object. For the three other cases ($\mathbf{T}_{\mathrm{right}}$,$\mathbf{T}_{\mathrm{up}}$,$\mathbf{T}_{\mathrm{down}}$) the matrix looks slightly different but the general structure is the same. If $p_{\mathrm{suc}} < 1$, a correct choice may be erroneously considered as unsuccessful and therefore not be chosen again. This leads to a case where all four choices have been taken and all are recognised to be ‘incorrect’. In that situation the model will reveal that something went wrong and start over, meaning going back to state s_1 (rows 12, 13 and 14 in $\mathbf{T}_{\mathrm{left}}$ (1)).

The next parameter of the model represents a subject’s ability to remember its previous actions on an object. As we use a Markov model, the history of actions is encoded in the state of the model, such that the probability a subject remembers the current state represents this ability. We define p_{sr} as *state recall probability*, which implies that $1 - p_{\mathrm{sr}}$ represents the probability to confuse the current state with all other possible states. The success state s_{16} should not be confused. Thus p_{sr} is located between an ideal learner ($p_{\mathrm{sr}} = 1$) and a total confused one ($p_{\mathrm{sr}} = 1/15$). We collect these probabilities in the state confusion matrix $\mathbf{C}$ with the ijth element being the probability of assigning state s_i to s_j. The subject’s attention is uniformly distributed over all states. In case of an ideal learner $\mathbf{C}$ becomes the identity matrix.

$$\mathbf{C} = \begin{pmatrix} p_{\mathrm{sr}} & \frac{1-p_{\mathrm{sr}}}{14} & \frac{1-p_{\mathrm{sr}}}{14} & \cdots & \frac{1-p_{\mathrm{sr}}}{14} & 0 \\ \frac{1-p_{\mathrm{sr}}}{14} & p_{\mathrm{sr}} & \frac{1-p_{\mathrm{sr}}}{14} & \cdots & \frac{1-p_{\mathrm{sr}}}{14} & 0 \\ \vdots & & \ddots & & \vdots & \vdots \\ \frac{1-p_{\mathrm{sr}}}{14} & \cdots & \frac{1-p_{\mathrm{sr}}}{14} & p_{\mathrm{sr}} & \frac{1-p_{\mathrm{sr}}}{14} & 0 \\ \frac{1-p_{\mathrm{sr}}}{14} & \cdots & \frac{1-p_{\mathrm{sr}}}{14} & \frac{1-p_{\mathrm{sr}}}{14} & p_{\mathrm{sr}} & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

As a subject may confuse the shown objects among each other, we introduce a third parameter. The *object recall probability* p_{or} is the subject's chance to remember the current object. In reverse, $1 - p_{\mathrm{or}}$ is the probability to confuse the current object with all other possible objects. A series containing eight different objects leads to $1/8 \leq p_{\mathrm{or}} \leq 1$. Here $p_{\mathrm{or}} = 1$ represents a perfect recall of objects and $p_{\mathrm{or}} = 1/8$ describes a learner without any memory for objects. We define the object confusion matrix $\mathbf{O}$ with the ijth element being the probability of assigning object i to object j. Again, the attention is uniformly distributed over all objects and, as above, in case of an ideal learner $\mathbf{O}$ becomes the identity matrix.

$$\mathbf{O} = \begin{pmatrix} p_{\mathrm{or}} & \frac{1-p_{\mathrm{or}}}{7} & \frac{1-p_{\mathrm{or}}}{7} & \cdots & \frac{1-p_{\mathrm{or}}}{7} \\ \frac{1-p_{\mathrm{or}}}{7} & p_{\mathrm{or}} & \frac{1-p_{\mathrm{or}}}{7} & \cdots & \frac{1-p_{\mathrm{or}}}{7} \\ \vdots & & \ddots & & \vdots \\ \frac{1-p_{\mathrm{or}}}{7} & \cdots & \frac{1-p_{\mathrm{or}}}{7} & p_{\mathrm{or}} & \frac{1-p_{\mathrm{or}}}{7} \\ \frac{1-p_{\mathrm{or}}}{7} & \cdots & \frac{1-p_{\mathrm{or}}}{7} & \frac{1-p_{\mathrm{or}}}{7} & p_{\mathrm{or}} \end{pmatrix} \tag{3}$$

With the defined state space and transition probabilities the learning process for every single object can be described. For each object the model performs a trial by selection of a button, following the state transition matrix. As this is not a deterministic process we calculate the probabilities to be in a certain state $\boldsymbol{\pi} = P(s_i)$ for all $s_i \in S$. Always starting in state s_1, the initial distribution $\boldsymbol{\pi}(t=0)$ is $\pi_1(t=0) = 1$ and $\pi_{2\ldots16}(t=0) = 0$. For every trial the new state probability vector $\boldsymbol{\pi}(t+1)$ results from the product of the transposed transition matrix $\mathbf{T} \in \{\mathbf{T}_{\mathrm{left}}, \mathbf{T}_{\mathrm{right}}, \mathbf{T}_{\mathrm{up}}, \mathbf{T}_{\mathrm{down}}\}$ and the previous state probability vector $\boldsymbol{\pi}(t)$. Since we allow state confusion, the state confusion matrix gets also part of the product

$$\boldsymbol{\pi}(t+1) = \mathbf{T}^{\mathrm{T}} \cdot \mathbf{C} \cdot \boldsymbol{\pi}(t) \ . \tag{4}$$

After every trial we need to sum up the actual probability of success and the probability of success of the previous trial. For instance, if we consider 'left' as the correct button, the state s_{15} automatically leads to success (s_{16}) in the next trial (cf. Tab. 1 and (1)). Thus, this sum becomes $\pi_{15}(t+1) = \pi_{15}(t) + \pi_{16}(t)$ for that case.

We are now able to calculate the state probabilities for each object separately. So far the learning process is represented by the state probability vectors $\boldsymbol{\pi}^l$ for each object l of the series. As we consider the chance to confuse objects we have to update the state probabilities for all objects, even if just one is really shown to the subject. This implies, that the probability of success on a presented object is the weighted sum of the success states of all objects with their probability of appearance. A series of eight objects $l = 1, \ldots, 8$ yields the probability of success P_k for object k

$$P_k(t) = \sum_{l=1}^{8} O_{kl} \cdot {\pi^l}_{16}(t) \qquad k \in \{1, \ldots, 8\}. \tag{5}$$

So, P_k is the final probability the model picks the correct button if object k is shown.

Before the next learning step, we weight the state probability vectors of all objects $\boldsymbol{\pi}^l(t)$ with the probability to confuse the current object k with object l

$${\boldsymbol{\pi}^l}_{\text{update}} = O_{kl}\boldsymbol{\pi}^l + (1 - O_{kl})\boldsymbol{\pi}^l \quad l = 1, \ldots, 8 \ \text{ and } \ k \ \in \{1, \ldots, 8\} \ . \tag{6}$$

We have to consider the chance that the current object k already was confused, which is the element O_{kl} for 'no confusion' and $1 - O_{kl}$ for 'confusion'. This ensures that all state probability vectors are updated even if just object k is shown since object k might be confused with all other objects. In case of a perfect recall of objects this step does not change the state probability vectors since $O_{kl} = 0$ if $k \neq l$ and $O_{kl} = 1$ if $k = l$.

Figure 2 shows the workflow of the model. Briefly summarised, for every single object we model a learning process through 16 possible states. The state space is given by the experimental set up. The state transitions are given by the assumed policy a subject follows. Further we added three parameters to the model. Each controls a typical mistake that might occur during trial and error learning in the experiment. These parameters are: (i) the success recall probability p_{suc} giving the chance to confuse 'success' and 'failure', (ii) the state recall probability p_{sr} representing the models ability to remember its previous actions on an object and (iii) the object recall probability p_{or} indicating the chance to remember the current object in a later trial.

Figure 3 shows the learning curves of the model for each object in a series of eight. In this simulation eight objects are shown to the model. The objects are presented for ten cycles. That is, every single object is shown once per cycle,

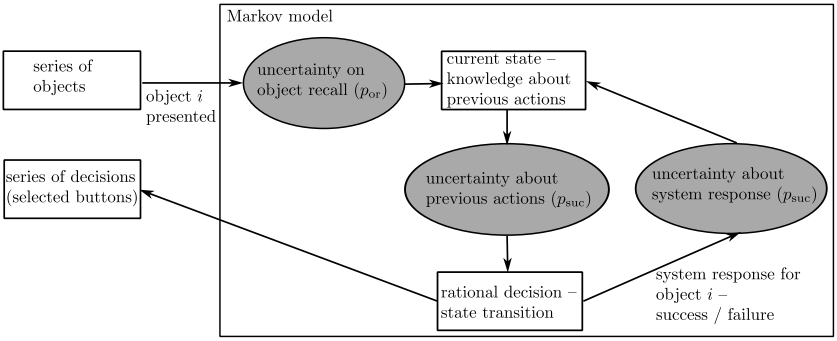

Markov model
series of objects
object i presented
uncertainty on object recall (p_{or})
current state – knowledge about previous actions
uncertainty about previous actions (p_{suc})
uncertainty about system response (p_{suc})
series of decisions (selected buttons)
rational decision – state transition
system response for object i – success / failure

Fig. 2. Workflow of the Markov model

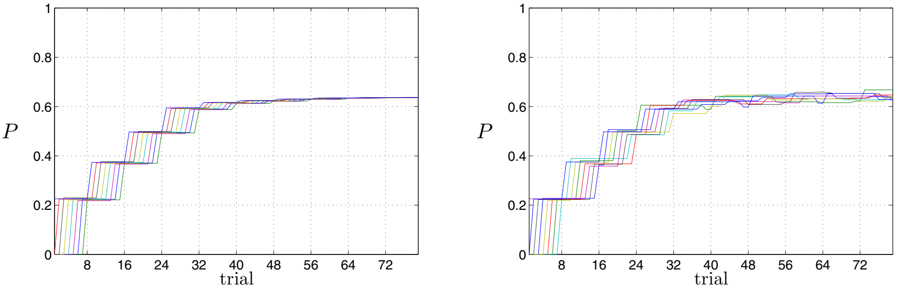


Fig. 3. Learning process of the Markov model: deterministic order (left) and random order (right). The parameters were $p_{\text{suc}} = p_{\text{sr}} = p_{\text{or}} = 0.9$. Every line shows the probability of success P (cf. 5) for the concerning object.

which results in ten trials on every object and 80 trials in total. Further we distinguish a deterministic and a random order. In case of a deterministic order the objects are presented in the same succession in every cycle. For the random case the succession of the objects is randomised in every cycle.

3 Analysis of the Markov Model

The advantage of a Markov model is the possibility to an analytic view on the learning process. The following analysis is limited on the explicit associative learning task (Fig. 1) on a *single* object. This is the model as it is described above until (4). The steady state and convergence of the Markov chain illustrate the correlation of the parameters p_{suc}, p_{sr} and their influence during learning. We gain no further finding on the role of the parameter p_{or} by this analysis, since this parameter is influential on the learning process for a *series* of objects.

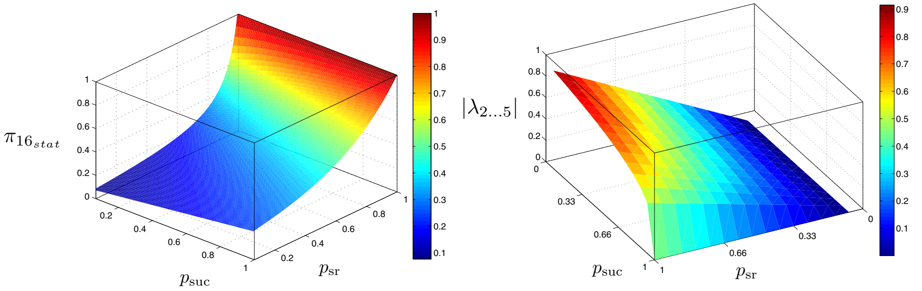
The steady state has its relevance in the asymptotic character of the state probability vector $\boldsymbol{\pi}(t)$ for $t \to \infty$. For convenience, we shorten $\mathbf{T}^{\mathrm{T}} \cdot \mathbf{C} = \mathbf{A}$ in (4). Repetitive replacement in (4) of $\boldsymbol{\pi}(t+1)$ by $\mathbf{A} \cdot \boldsymbol{\pi}(t)$ gives the steady state:

$$\boldsymbol{\pi}_{stat} = \lim_{t\to\infty} \mathbf{A}^t \boldsymbol{\pi}(0) \qquad \hat{=} \qquad \boldsymbol{\pi}_{stat} = \mathbf{A}\boldsymbol{\pi}_{stat} \ , \tag{7}$$

leading to the linear system of equations

$$\pi_{j_{stat}} = \sum_i A_{ij} \pi_{i_{stat}} \ , \quad i,j \in S \ . \tag{8}$$

With the analytic solution for the steady state (8), we are able to calculate the final state probability vector of the Markov chain. A closed solution can be found with a computer algebra system. The final probability of success $\pi_{16_{stat}}$ depends only on p_{suc} and p_{sr} in a fraction which numerator is a sixth-degree polynomial of p_{sr} and second-degree of p_{suc}. The denominator is a fourth-degree polynomial of p_{sr} and first order for p_{suc}. The whole term is to long to be shown here and its structure does not provide any extra information. Nevertheless, it represents the probability of success on the explicit task of associating an object with the rewarded choice if the object is shown in an infinite loop. So, $\pi 16_{stat}$ is the upper limit for the performance of the model and can be calculated solely based on the parameters p_{suc} and p_{sr}. Figure 4(a) shows $\pi 16_{stat}$ for variable p_{suc} and p_{sr}. The impact of p_{sr} is much greater than p_{suc}. If $p_{\mathrm{sr}} < 1$ than p_{suc} only slightly affects $\pi 16_{stat}$.



(a) (b)

Fig. 4. Influence of p_{suc} and p_{sr} on the learning process for a *single* object. (a) upper limit of the success probability ($\pi 16_{stat}$). (b) absolute value of the eigenvalues ($|\lambda_{2\ldots5}|$).

The convergence of the learning process is also affected by the parameters. $\mathbf{A} = \mathbf{T}^{\mathrm{T}} \cdot \mathbf{C}$ has five non-zero eigenvalues whereas λ_1 is dominant and $\lambda_{2\ldots5}$ take the form

$$\begin{aligned}
\lambda_1 &= 1 \\
\lambda_2 &= \tfrac{1}{28}(12 - 12p_{\mathrm{suc}})^{\frac{1}{4}} \cdot (15p_{\mathrm{sr}} - 1) \\
\lambda_3 &= -\tfrac{1}{28}(12 - 12p_{\mathrm{suc}})^{\frac{1}{4}} \cdot (15p_{\mathrm{sr}} - 1) \\
\lambda_4 &= I\tfrac{1}{28}(12 - 12p_{\mathrm{suc}})^{\frac{1}{4}} \cdot (15p_{\mathrm{sr}} - 1) \\
\lambda_5 &= -I\tfrac{1}{28}(12 - 12p_{\mathrm{suc}})^{\frac{1}{4}} \cdot (15p_{\mathrm{sr}} - 1).
\end{aligned} \tag{9}$$

For reasonable combinations of p_{suc}, p_{sr} follows

$$|\lambda_{2\ldots5}| < 1 \qquad \text{if} \qquad \frac{1}{15} \leq p_{\mathrm{suc}}, p_{\mathrm{sr}} \leq 1. \tag{10}$$

The state probability vector $\boldsymbol{\pi}$ can be written in terms of its orthogonal basis

$$\boldsymbol{\pi} = \sum_{i=1}^{16} \frac{\langle \boldsymbol{\pi}, \boldsymbol{e}_i \rangle}{|\boldsymbol{e}_i|^2} \boldsymbol{e}_i, \tag{11}$$

such that (7) at time t is

$$\boldsymbol{\pi}(t) = \mathbf{A}^t \boldsymbol{\pi}(0) = \sum_{i=1}^{16} \frac{\langle \boldsymbol{\pi}(0), \boldsymbol{e}_i \rangle}{|\boldsymbol{e}_i|^2} \mathbf{A}^t \boldsymbol{e}_i \tag{12}$$

which is by the definition of the eigenvalues

$$\boldsymbol{\pi}(t) = \sum_{i=1}^{16} \frac{\langle \boldsymbol{\pi}(0), \boldsymbol{e}_i \rangle}{|\boldsymbol{e}_i|^2} \lambda_i^t \boldsymbol{e}_i . \tag{13}$$

Since λ_i is taken to the power of t, the contribution of the eigenvalues $\lambda_{2\ldots5}$ to the sum in (13) is lowered with every step ($|\lambda_{2\ldots5}| < 1$ (10)). The smaller they are the faster they disappear in the sum and the faster the sum converges to the steady state. Figure 4(b) shows $|\lambda_{2\ldots5}|$ against the parameters p_{suc} and p_{sr}. The impact of p_{suc} is stronger than those of p_{sr}.

4 Results and Discussion

We fitted the Markov model on experimental data published in [4]. Further, a similar experiment is presented in [5]. The subjects were tested on the random and deterministic ordered sequence as described above. The experiment lasts for 28 cycles. The three model parameters were manually selected to fit the subjects' learning behaviour (Fig. 5). Yet, we did not use an optimisation procedure to gain better fits and/or other parameter combinations.

Compared to the reinforcement learning model in [5] and the neural network model in [3], the Markov model ignores temporal context. Therefore, it makes no predictions concerning the performance difference which occurred in the experiment for different sequential orders of object presentation. Nevertheless, it gives the chance to an analysis of the underlying learning task (object association with the correct button) which was not possible with the existent models in such explicit way.

The associative learning of a single object is described with 16 possible states. The assumed rational behaviour gives the state transitions. The parameters (p_{suc}, p_{sr}, p_{or}) model typical mistakes. The analysis of the proposed Markov chain produced two main results.

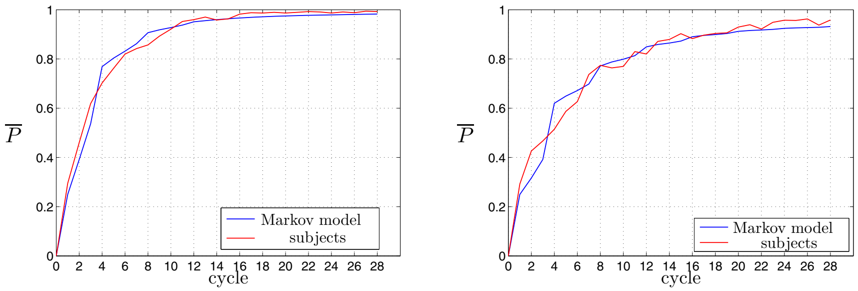


Fig. 5. Learning process of the basic Markov model on a deterministic (left) and a random sequence (right). The parameters were $p_{\text{suc}} = 0.6$, $p_{\text{sr}} = 1$, $p_{\text{or}} = 0.9875$ (left) and $p_{\text{suc}} = 0.3$, $p_{\text{sr}} = 1$, $p_{\text{or}} = 0.9875$ (right). The mean probability of success $\overline{P}$ is averaged over all subjects.

First, the probability, a subject memorises a success correct (p_{suc}), affects mainly the slope of the learning curve (cf. Fig 4(b)). Second, the probability a subject remembers the learning history for a certain object (p_{sr}) is primarily responsible for the long term development of the success probability (cf. Fig. 4(a)).

We see the relevance of this kind of model first and foremost in the design process of future experiments. The parameters can be influenced directly through the test arrangement. For instance, one could use other objects than fractals to ease the memorization which is reflected in the object recall probability (p_{or}). Or, one could change the way the reward is given, which should affect the success recall probability (p_{suc}). In that way the model could help to get a first impression of the experimental results to be expected.

References

1. Bunge, S.A., Wallis, J.D., Parker, A., Brass, M., Crone, E.A., Hoshi, E., Sakai, K.: Neural circuitry underlying rule use in humans and nonhuman primates. J. Neurosci. 25(45), 10347–10350 (2005)
2. Gaffan, D., Harrison, S.: Inferotemporal-frontal disconnection and fornix transection in visuomotor conditional learning by monkeys. Behav. Brain Res. 31, 149–163 (1988)
3. Glüge, S., Hamid, O.H., Wendemuth, A.: A Simple Recurrent Network for Implicit Learning of Temporal Sequences. Cognitive Computation 2(4), 265–271 (2010)
4. Hamid, O.H., Braun, J.: Task-irrelevant temporal order and learning of arbitrary visuo-motor associations. In: Perception 36: ECVP 2007, Arezzo, pp. 50–51 (2007)
5. Hamid, O.H., Wendemuth, A., Braun, J.: Temporal context and conditional associative learning. BMC Neuroscience 11, 45 (2010)
6. Miyashita, Y.: Neuronal correlate of visual associative long-term memory in the primate temporal cortex. Nature 335, 817–820 (1988)
7. Miyashita, Y., Chang, H.S.: Neuronal correlate of pictorial short-term memory in the primate temporal cortex. Nature 331, 68–70 (1988)
8. Muhammad, R., Wallis, J.D., Miller, E.K.: A comparison of abstract rules in the prefrontal cortex, premotor cortex, the inferior temporal cortex and the striatum. J. Cognit. Neurosci. 18, 974–989 (2006)
9. Petrides, M.: Conditional learning and the primate frontal cortex. In: Perecman, E. (ed.) The Frontal Lobes Revisited, pp. 91–108. The IRBN Press, New York (1987)
10. Wise, S.P., Murray, E.A.: Arbitrary associations between antecedents and actions. Trends Neurosci. 23(6), 271–276 (2000)

General Theory of Exobehaviours: A New Proposal to Unify Behaviors

Sergio Miguel Tomé[*]

Universidad de Salamanca
sergiom@usal.es

Abstract. Nowadays science has not found a way to unify the behavior of biological and autonomous nonbiological systems. While psychology uses the property of intelligence as a basis for explaining cognitive behaviors, artificial intelligence has been unable to explain that property and provide to nonbiological systems with it. In addition, discoveries in the last decade have demonstrated the existence of random and cyclic behaviors in nature that complicate the possibility of unifying the behaviors that are known so far in living organisms. This article presents a new proposal, called general theory of exobehavior, to explain behaviors in a unified way of biological and autonomous non-biological systems, and achieve a foundation for AI as a science.

Keywords: behaviors, unification, general theory of exobehavior.

1 Introduction

Science tries to explain sets of phenomena of nature through the scientific method. One of the sets that it is trying to explain is the behaviors of living beings. The research to achieve an explanation for the behavior of living things has been very extensive, and it even generated new scientific disciplines. During research, science has sought to clarify the physiological mechanisms that generate behaviors as well as theories capable of predict them. Some of the findings and studies that have led to the current state on the issue are briefly exposed here. In the 18th century, Luigi Galvani's investigations led him to propose that animal brains generate electricity to move the muscles. It is also important to say that in 1727 the French astronomer de Mairan reported that the movement of leaves in some plants arose from an endogenous rhythm. In the 19th century, it is worth noting Franz Joseph Gall's proposal that the mental functions of humans were found in specific brain areas, which was proved in 1861 by Paul Broca when he located the speech center in the brain. Returning to more general works, the investigations of Ivan Mikhaylovich Sechenov on reflexes and brain were central to Ivan Pavlov, who later made an experimentally proven theory of conditioned reflexes.

[*] I wish to thank Dr. Luis Alonso Romero for his support to my work in the line of research that is presented in this article and his comments, and Dr. Rodolfo Llinás for our long and valuable discussions about intelligence.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 20–29, 2011.

At the end of the century, Ramón y Cajal revolutionized research on the nervous system with his neuronal theory and the law of the depolarization, which made him the father of a new field of science: neuroscience. In the 20th century, the idea that problem solving by humans (cognitive behavior) is caused by a property called intelligence is consolidated in the scientific community due to Alfred Binet's works about its measurement and further research by Charles Spearman and Thurstone Louis. Regarding animal behavior, Edward Thorndike, through his research, enunciated the law of effect in 1911. Thorndike's work was the basis for the later formulation and development of operant conditioning by Burrhus Frederic Skinner. In 1926, Walter Cannon took up the concept of Claude Bernard's internal environment and proposed a property of the systems that regulates its internal environment, homeostasis. Other important contributions during the 20th century were the works of Konrad Lorenz and Tinbergen Nikolaas explaining instinctive behaviors through what is now known as fixed action patterns. In 1943, a new approach to try to explain the generation of behaviors is made in the article by Warren McCulloch and Walter Pitts. In the article, they argued that the activities of neurons and the relationship between them could be studied mathematically. On the other hand, in 1948 Edward Chace Tolman proposed cognitive maps. Tolman's ideas allow the explanation of complex behavior as observed by Wolfgang Köhler in chimpanzees that was not explainable by the law of effect, and paves the way for the cognitive revolution that would happen in the 50's. A year later Donald Hebb formulated his theory of cell assembly combining psychology and physiology.

The mentioned investigations caused that, during the second half of the 20th century, science established that from a functional point of view the behavior of living things is divided into seven groups: homeostatic behaviors, biological rhythms, reflexes, fixed action patterns, classically conditioned reflexes, operantly conditioned reflexes, and cognitive behaviors.

Homeostasis has been the framework to explain cellular behavior. Biological rhythms have been studied over the past centuries, but only with the recent development of molecular biology, it began to provide answers to their issues. Concerning the following four groups of behavior, neuroscience has achieved to explain several of them at the neural level; but it is still working on others, especially in cognitive behaviors. Therefore, the property of intelligence remains as a concept used in explaining cognitive behaviors. But even when neuroscience will achieve to explain cognitive behaviors, the issue will not be closed. In the last decade there have been numerous discoveries that have completely changed the outlook on behaviors. One of the most important discoveries has been the behavior of bacteria that can not be explained with the framework of homeostasis[1][2], and is even referred to as prokaryotic intelligence[3]. In addition, random behaviors which allow greater survival than sensing behaviors have been discovered[4][5]. And it has also been discovered that some cyclic behaviors fit better than sensing behaviors in some environments[6][7].

Another issue that science has dealt with behavior has been the realization of these by non-biological systems. The advent of computers in the 20th century, made that, on the one hand, mathematicians and engineers set the goal of providing intelligence to machines, while on the other hand, the cognitive psychologists' opinion was that computers were perfect elements to test their theories about information processing by the brain. Thus the artificial intelligence (AI) appeared in 1956. However, in the discipline of AI, despite successes such as engineering, there is a double failure: it has been impossible to base the AI as a science, nor to achieve the goal of a machine possessing general intelligence and displaying the cognitive behaviors of human beings. The inability to get a machine with the cognitive behavior of a human being after a great effort by the scientific community has opened a debate on the need for new concepts[8][9] and it has been exposed that intelligence is a pre-scientific term that should be left to advance[8].

Taking into account the findings and investigations that have been made by the scientific community about the behaviors of biological systems and the generation of behavior by non-biological systems, the following questions are perceived.

1. Neurobiology explains the behavior of living creatures with a nervous system using concepts such as action potentials, neurotransmitters, synapses, or electrophysiological properties[10]. The behavior of plants and bacteria is explained by concepts such as gene, protein or promoter[1][2]. In the case of computers, the symbolic paradigm uses the concepts of inference engine, entropy or matching algorithm among others[11]. Is it possible to find a scientific language that covers all sets of behaviors?
2. One of the goals of science is to unify sets of phenomena. Today, science deals with the behavior of living things dividing by them into different sets. Furthermore, recent research has shown effective behaviors in nature that had not been previously considered as the random and cyclic. Is it possible to find a physical principle that unifies all these sets?
3. After more than 50 years, AI has been developed as knowledge engineering, but it does not achieve a foundation as science[8]. Research has shown that the property of the intelligence is not a useful concept for addressing the issue of cognitive behavior in autonomous nonbiological systems, but is it possible to find a concept that unifies behaviors of biological and nonbiological autonomous systems and support IA as a science?

This paper presents a theoretical proposal called the General Theory of Exobehaviour (GTE), which addresses the three issues mentioned above. The GTE is a proposal that has been developed following the method of reasoning about nature developed by Newton, and meets the criteria of monism, physicalism and reductionism. The organization of the paper is as follows. Section two defines all behavior as a single whole. In the third section, a physical property is proposed as the cause of behavior following the Newton's method and it is interpreted in physical terms. The fourth section presents a set of laws governing phenomena of populations based on the proposed property. The last section contains a discussion on the proposal made in this paper and some conclusions.

2 A Physicalist Classification of Behaviors

The first element of the GTE is a physicalist definition of behaviors. By observing the behavior of living beings and autonomous non-biological systems, one finds: motor behaviors (e.g. movement of animals, robots, or plant leaves), luminescent behaviors (e.g. firefly), active mimicry behaviors (e.g. Chameleon), electrical behaviors (e.g. Electric eel), behaviors of excretion (e.g. octopus) and change of phenotype (e.g. Escherichia coli). In all these behaviors there are three characteristics:

1. There is a energy source capable of generating a continuous flow of energy that either works on a feature of the system, or leaves the system and affects the environment.
2. The energy source is not regulated from outside the system. That is, nothing from outside the system can directly induce, graduate or stop the source of energy.
3. The energy flow is not directed from outside the system. So, nothing from outside the system is capable of choosing the way of the energy within the system.

Using the above characteristics it will be defined the type of phenomenon that will be explained by the GTE, which will be called exobehavior.

Definition 1 (Exobehavior of a system). *When a system has energy flows that are not directed from outside and the energy sources of these flows are not regulated also from outside, the sequence that is generated to order temporally all the values of energy flows that acts on its characteristics will be called the exobehavior of the system.*

In the definition of exobehavior has to be noted that an exobehavior lasts as long as the system has energy (e.g. in the case of a living being throughout all his life).This differs from what is commonly understood by behavior, because in psychology, or ethology, behavior is a very specific sequence of actions beginning with the introduction of a living being in an environment, or when submitting a new stimulus to it, and lasts as long as the living being stays in the environment, or while its response to new stimuli lasts.

Since the GTE is a theory that covers biological and nonbiological autonomous systems, it does not use the term “live”, which it is replaced by “exoactive”. A system is *exoactive* when it performs an exobehavior.

It should be noted that the energy flow at the macroscopic level may be continuous or discontinuous. If the system has a continuous flow of energy it can always carry out its behavior; if it is discontinuous it can not. For this reason, if the flow of energy is not continuous system behavior is called *semiexobehavior* since when energy flow is interrupted the behavior of the system does not depend on itself. In the case of the flow of energy being directed externally but not regulated, or inverse, the behavior of the system is called *subexobehabiour*.

The goodness of this new classification is that it manages to have a single set of phenomena, exobehaviors, which contains the behaviors of biological systems, ranging from the movement of a bacterium to the human speaking and the behaviors of autonomous non-biological systems such as robots and computers, defined by physical concepts.

3 The Cause of Exobehaviors

Once defined a set of phenomena is necessary to address an explanation.The reasoning method developed by Newton[12], in which the scientific explanation is based, considers that the cause of a set of phenomena is a property of matter. Therefore, the main hypothesis of the GTE is the existence of a property in the matter that causes exobehaviors, which will be called *fas*.

3.1 The Microscopic Interpretation of Fas

Following Newton's method, the existence of a physical property to explain exobehaviors has been proposed. But Newton's method does not explain anything more about the property. However, with current knowledge of the matter a microscopic interpretation is required. The interpretation is as follows.

The point of view of computer engineers and computer scientists is that a physical system is potentially a computer. This point of view shall be called "potential computation" and is based on the next idea. Any element of matter that has at least one degree of freedom, so it has several distinguishable states, is capable of storing information. However, there is another view on the computation that is in the physics of information. The physics of information was born more than 50 years ago, but only in the last two decades has received great interest. The physics of information has shown that information and computation have a physical character. One of the cornerstones of this branch of physics has been the Landauer principle[13] that says that erasing one bit of information has $k_b\ ln2$ of thermodynamic entropy. The physics of information considers that the universe is a computer that is constantly computing[14]. I will call this other point of view on computation "actual computation". The idea behind this view is that any element of matter that has at least one degree of freedom is storing information. Thus, each energy process that modifies the state of one element by another is considered a computational operation. Therefore, from the standpoint of actual computer, every physical system, whether biological or not, is carrying out computation constantly.

The Newton method of reasoning dictates that exobehaviors are generated by a property, fas. Using the concept of actual computation of the physics of information, fas must be interpreted microscopically as the computational process that occurs in the system and controls the energy flows of the system. However, the existence of a microscopic interpretation does not imply that the calculations must be done at that level. For example, thermodynamic temperature has a microscopic interpretation based on mechanics, but it is not feasible to predict the future state of a system at macroscopic level through mechanical calculations in the microscopic level due to the number of particles. The order of magnitude of particles in a macroscopic system is 10^{23}.

On the other hand, the microscopic interpretation of fas makes compatible the idea of a property of the matter which is proposed in the GTE with the hypothesis which have been used by cognitive psychology and AI in his research that mind is a computational process.

3.2 The Mathematical Description of Fas

The next question is how to describe the values of the fas. The mathematical framework to describe the values of the fas must comply with certain requirements. The first one is that it must be independent of the computational mechanism. The reason for this requirement is that an animal has a neural network, a bacterium has a biochemical network and a computer has a finite record architecture, but each of its fas values should be described in the same mathematical framework. The mathematical framework that satisfies the requirement comes from the works to explore the limits of computable. Turing found a limit of computable, but the limit found belonged to Turing machines. Thus, it was needed to know if the Turing limit was the same regardless of the mechanism of computation. For this purpose, a functional approach was applied. Therefore, the functional approach is the appropriate mathematical framework, and a value of fas be described using a function. It is important to note that a value of fas, due to the microscopic interpretation of the fas, can store information which could change its response to the future. Therefore, the functions are of computational type. Another important requirement is that the mathematical framework must reflect each computational operation which is made by a system. Thus, the functions must be intensional[15]. According to this, the general equation of the GTE is the following functional equation.

$$\varepsilon_s(t) = \rho f_s(p_0, ..., p_m) \qquad m \geq 0$$

- $\varepsilon_s(t)$ are the values of the exobehaviour of the system-s in the moment t.
- ρ is the relation between fas and the exobehaviour of a system.
- $f_s(p_0, ..., p_m)$ is the function that describe the fas value of the system-s.
- $p_0, ..., p_m$ are parameters.

Thus, in the GTE, exobehaviors and fas are related by the above equation. So a value of fas is described by a function and every function that describes a value fas has its own parameters.

4 Postulates

Psychology and AI have been investigating in creating theories to predict and generate behavior, but their investigation ends in achieving predict or generate behavior. However, behaviors have important effects on nature. Whether individuals of a population have a behavior or other, it has important effects on the population. These effects are studied by evolutionary biology. Already in the 19th century, Francis Galton proposed that the intellectual capacities of human beings were innate and they were subjected to natural selection as well as the physical characters. Therefore, a theory of behavior must also explain the evolution of behavior in nature, as the evolution theory explains the evolution of the physical characteristics of living beings. Thus, this section set forth five postulates that relate fas with phenomena of populations.

Postulate 1 (Principle of fas fundamental kinds). *There are three fundamental kinds of fas that can generate an exobehavior: sequential, random and sensitive. Each value of fas belongs to one of them, or is a combination of them.*

The definition of each one of the named types are as follows. The fas of a system is sequential when each value of the exobehavior is in a sequence. The fas of a system is random when every value of the exobehavior is obtained in function of a random variable that is independent of the macroscopic state of the system environment. The fas of a system is sensitive when each value of the exobehavior is obtained in function of some macroscopic information of the system environment. It can easily be shown that each kind of fas generates exobehaviour. One can create for every kind of fas, a computer program that generates movements for a mobile robot using the definition of every kind of fas.

Postulate 2 (Law of the exoactivity time of a system). *In an environment with exoactivity restrictions where there are two systems with identical physical characteristics but different fitness of their fas values, the system whose fas value has higher fitness remains exoactive longer than the one with lower fitness. It occurs with a frequency strictly higher instead of the contrary.*

The existence of exoactivity restrictions means that there are states of the environment in which the system is exoinactive and hence ends its exobehavior. The fitness of a fas value to an environment is the ability to prevent the states where the system is not exoactive. This is a kind of natural selection when the systems are biological, and so this law links the GTE with the theory of evolution.

Postulate 3 (Law of the sequential fas). *In an environment with exoactivity restrictions where the state of the environment changes instantly and its states shape a stable homogeneous sequence, there is a sequential fas value that if it is synchronized with the stable homogeneous sequence it gives a higher average time of exoactivity to a population than whether they would have any sensitive fas value or random fas value.*

A state of an environment is homogeneous if all points of space have the same characteristics. A stable homogeneous sequence is when it has fluctuations which cause a heterogeneous state but it regains quickly to a homogeneous state. The average time of exoactivity of a population is the average of the exoactivity time of each one of its individuals.

Postulate 4 (Law of the random fas). *In an environment with exoactivity restrictions where each state is homogeneous and the state of the environment changes randomly and instantly, if a population have the random fas value that is adjusted to the probabilities of the environment and the number of its individuals exceeds the number of states of the environment, this population has a higher average time of exoactivity than whether they would have any sensitive fas value or sequential fas value.*

Postulate 5 (Law of the sensitive fas). *In an environment with exoactivity restrictions where each state is heterogeneous and the state of the environment changes in function of the previous environment state in a determinist way, there is a sensitive fas value with which a population has a higher average time of exoactivity than whether they would have any random fas value or sequential fas value.*

A state of an environment is heterogeneous if there is at least a space point that has different characteristics to the rest.

All postulates that have been stated are falsifiable. The first would be false if one found a kind of fas different of the kinds stated in the GTE, what is equivalent to find a kind of computational function different of the three kinds used in the GTE. The rest of postulates are hard to test with populations of living beings, but can be tested with experiments in virtual worlds, since the GTE covers nobiological autonomous systems. Thus, the GTE is a scientific theory. Also, the last three postulates, which have been stated, are supported by some biological evidences. The third postulate is supported by [7][6]; the fourth postulate is supported by [4] [5]; the fifth postulate is supported by [1][2].

5 Discussion and Conclusions

Since the birth of AI, a theory to explain the behavior of biological and autonomous nonbiological systems together has been sought and numerous proposals have been done, but the goal has not yet been achieved. The theory proposed in this paper is only the basis for developing a theory that achieves this unification.The proposal made has several difficulties which must be taken into account. The first issue is that the GTE is a scientific theory, and therefore, one of the following steps in the GTE must be making experiments to test the postulates. Another difficulty to consider is that it is not possible to know if the GTE has more postulates other than those stated here.

Another important issue is that the proposal submitted does not explain any particular behavior, since the theory states a fundamental equation whose unknown factor is an intensional computational function. So, until there is an intensional computing function that generates an exobehavior there is not a precise explanation for this. Thus, GTE will achieve explain completely behaviors only if all the solutions of each exobehavior equation are found, but it presents two problems: there is no mechanical method to do it and the number of functions could be very high.

In addition to the above difficulties, the reader may find hard to accept the idea of a property of matter to explain behaviors and the double macroscopic/microscopic interpretation. But the situation is similar to that of thermodynamics, where there is a property when the system is observed at the macroscopic level, the temperature, which has a microscopic interpretation, and there are laws that govern the interactions between systems based on the property. Moreover, intelligence, although we are used to the term, is also a property that is considered the cause of cognitive behavior.

Despite the above difficulties, the GTE has important differences compared to the proposals made so far to explain behavior and give a foundation to AI, and that makes it very interesting. One of the differences of the new proposal against any of the unified theories of cognition and its corresponding cognitive architecture [16] [17] is that under the same principles GTE covers all the behaviors, while unified theories of cognition only deal with cognitive behavior. Subsumption architecture [18] aims to include reflexes and cognitive behavior, but leaves out random and cyclical behavior researched in the last decade. Another difference from the TGE compared to the previous proposals and mathematical proposals, such as AIXI[19], Gödel machines[20], or Ortholattices [21], is that GTE is based on a physical property and its postulates are empirically falsifiable, so the GTE give a foundation to AI as science. Regarding proposals of physical character, such as Penrose[22] based on quantum physics, or Doyle [23] extending classical mechanics, the difference is that GTE is based on a different field of the other proposals, the physics of information. Also, the GTE does not break with cognitive psychology research, or AI research, since due to the microscopic interpretation of the fas GTE is a new framework where all can be looked from a new point of view (e.g. a cognitive architecture is considered an value of fas by GTE). Moreover, due to its microscopic interpretation and its consequences on populations, fas is a bridge between the world of atomic particles and the world of the theory of evolution, a bridge between physics and biology. Because of this, if the experiments support the theory it takes a step towards a better understanding of nature.

In summary, the GTE is a different proposal from those that have been done so far, and although the proposal is complex and has many difficulties to be developed, it also has many benefits. Therefore, it is necessary to research in the proposal, because an interesting horizon will be opened through it if the experiments support the theory.

References

1. Tagkopoulos, I., Liu, Y., Tavazoie, S.: Predictive behavior within microbial genetic networks. Science 320(5881), 1313–1317 (2008)
2. Mitchell, A., Romano, G.H., Groisman, B., Yona, A., Dekel, E., Kupiec, M., Dahan, O., Pilpel, Y.: Adaptive prediction of environmental changes by microorganisms. Nature 460(5881), 220–224 (2009)
3. Marijuán, P.C., Navarro, J., del Moral, R.: On prokaryotic intelligence: Strategies for sensing the environment. Biosystems 99(2), 94–103 (2010)

4. Beaumont, H., Gallie, J., Kost, C., Ferguson, G., Rainey, P.: Experimental evolution of bet hedging. Nature 462, 90–93 (2009)
5. Kussell, E., Leibler, S.: Phenotypic diversity, population growth, and information in fluctuating environments. Science 309(5473), 2075–2078 (2005)
6. McClung, C.R.: Plant circadian rhythms. Plant Cell 18(4), 792–803 (2006)
7. Green, R., Tingay, S., Wang, Z., Tobin, E.: Circadian rhythms confer a higher level of fitness to arabidopsis plants. Plant Physiology 129(2), 576–584 (2002)
8. Mira, J.M.: Symbols versus connections: 50 years of artificial intelligence. Neurocomputing 71(4-6), 671–680 (2008)
9. Brooks, R.A.: The relationship between matter and life. Nature, 409–411 (2001)
10. Hawkins, R., Abrams, T., Carew, T., Kandel, E.: A cellular mechanism of classical conditioning in aplysia: activity-dependent amplification of presynaptic facilitation. Science 219(4583), 400–405 (1983)
11. Schreiber, G.: Knowledge Engineering and Management: The CommonKADS Methodology. The MIT Press, New York (1999)
12. Newton, I.: Sir Isaac Newton's Mathematical Principles of Natural Philosophy and His System of the World. University of California Press, Berkeley (1934)
13. Landauer, R.: Information is physical. Physics Today 44(5), 23–29 (1991)
14. Lloyd, S.: Computational capacity of the universe. Phys. Rev. Lett. 88(23), 237901 (2002)
15. Church, A.: The Calculi of Lambda Conversion. Princeton University Press, Cambridge (1943)
16. Langley, P., Laird, J.E., Rogers, S.: Cognitive architectures: Research issues and challenges. Cognitive Systems Research 10(2), 141–160 (2009)
17. Newell, A.: Unified Theories of Cognition. Harvard University Press, Cambridge (1990)
18. Brooks, R.A.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation 2 (1), 14–23 (1986)
19. Hutter, M.: Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability. Springer, Berlin (2006)
20. Schmidhuber, J.: Gödel machines: self-referential universal problem solvers making provably optimal self-improvements. Tecnical Report IDSIA-19-03, pp. 256–264 (2003)
21. Trillas, E., Cubillo, S., Castiñeira, E.: On conjectures in orthocomplemented lattices. Artificial Intelligence 117(2), 255–275 (2000)
22. Penrose, R.: The Emperor's New Mind. Oxford University Press, Oxford (1989)
23. Doyle, J.: Extending Mechanics to Minds. Cambridge University Press, New York (2006)

Bio-inspired Decentralized Self-coordination Algorithms for Multi-heterogeneous Specialized Tasks Distribution in Multi-Robot Systems

Yadira Quiñonez, Javier de Lope, and Darío Maravall

Computational Cognitive Robotics
Dept. Artificial Intelligence
Universidad Politécnica de Madrid
ay.quinonez@alumnos.upm.es, javier.delope@upm.es, dmaravall@fi.upm.es

Abstract. This paper focuses on the general problem of coordinating multiple robots. More specifically, it addresses the self-election of heterogeneous specialized tasks by autonomous robots, as opposed to the usual multi-tasks allocation problem in multi-robot systems in which an external controller distributes the existing tasks among the individual robots. We are rather interested on decentralized solutions in which the robots themselves, autonomously and in an individual manner, select a particular task so that all the existing tasks are optimally distributed and executed. In this regard, we establish an experimental scenario and we propose a bio-inspired solution based on threshold models to solve the corresponding multi-tasks distribution problem. The paper ends with a critical discussion of the experimental results.

Keywords: Multi-robot systems, bio-inspired threshold models, multi-tasks distribution, self-coordination of multiple robots, multi-heterogeneous specialized tasks distribution.

1 Introduction

Autonomous Multi-robot Systems is one of the characteristic applied areas of Artificial Intelligence that has gotten an amazing growth since its inception until today, and it has developed very significant progress in various fields of application [1], becoming a fundamental tool to produce, work and perform dangerous jobs on earth and beyond. Currently, important research is being focused specifically to the coordination and communication of multi-robot systems, the interest in this field is that, these systems offer several advantages and a higher potential with respect to the use of a single robot in performing many tasks.

Some areas of application range from industrial to military, in all these areas, the tasks are very dangerous that require greater performance, coverage and robustness and fault tolerance. Some examples include: cooperation in the transport or handling of parts in the industry, the exploration of land wholly or partially unknown (e.g. an unknown planet exploration), tasks on large tracts

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 30–39, 2011.

of land, support in rescue operations and intervention in environment hostile or contaminated and surveillance operations, etc. [2],[3].

In general, a multi-robot system consists of a set of robots that, in the same environment, interact with each other to achieve a common goal [4], thus trying to improve the effectiveness, efficiency, performance and robustness of a robotic system. These systems provide greater flexibility in performing tasks and possible fault tolerance. To achieve that several robots coordinate with each other to perform a specific mission is not a trivial task, because, they must be designed to operate in dynamic environments in which we must also take into account the classical problems of autonomous robotics (e.g. uncertainty and unforeseen changes always present), new difficulties arising from the influence of the team robots on the environment and the task goal. For this reason, it is an issue that has aroused much interest to many researchers, both in the field of robotics and biology, in this regard; it is noteworthy that some biological approaches have served as inspiration for the design of coordination models in multi-robot systems.

For over many years, communities or colonies of social insects have been deeply studied by some researchers [5], [6], as they provide fascinating examples of functional collective behavior. From a biological standpoint, it has attracted much interest specifically for the way they carry out the division of labor, the fact that members of a colony specialize in certain roles. In this way, it takes place at the societal level, given a number of specific tasks, a division of labor and based on this assignment, individuals will acquire specific skills through practice. A colony of insects functions as an integrated unit that is capable of processing large amounts of information in a distributed manner. In the sequel we describe the so-called response threshold model for its particular interest as a theoretical model of how some social insects implement their social division of labor.

2 Related Work

The coordination among robots is an important aspect of research on multi-robot systems; there are usually two kinds of strategy for the coordination of multi-robots, which are centralized and distributed strategies. Currently, there are several studies that focus mainly on the coordination of a set of robots using different techniques, in order to solve a specific problem. Then, it described some potential trends of research articles related with the coordination of multi-agent systems, swarm robots and multi-robot systems.

A. Multi-agent systems

Price and Tiño suggest a number of strategies to address problems of task allocation in multi-agent systems, based on the principle of self-organization of social insects through the mathematical model developed by Bonabeau. They make a comparison of decentralized algorithms (FIFO and Greedy) to measure and evaluate the effectiveness of each strategy to process the mail and at the same time minimizing the number of changes [7]. The problem that has been considered for these algorithms of adaptation is a variation of the mail retrieval proposal by Bonabeau.

B. Swarm robots

Baglietto et al. have presented a coordination approach to swarm robots both navigation and task allocation based on RFID (Radio Frequency Identification, RFID). RFID devices are distributed a priori in the environment by building a navigation chart; each RFID device contains navigation instructions that allow the robots to run the routes from one place to another. Robots cannot communicate with each other, but may do so indirectly by writing and reading RFID devices. To perform the distributed task allocation algorithm defines an auction, where the central server takes work to be undertaken by a team of robots, analyzes and decides the number of robots, then robots are informed about the new tasks The allocation is the result of negotiations that each robot makes its own. Similarly using RFID devices to communicate, leaving registration messages between them, for example, messages and records assignments and out of zones. The system has been implemented in Player/Stage and navigation algorithm has been tested in MATLAB [8].

In the study by Yang et al. proposed a foraging mission in swarm robots using a task allocation algorithm based on the response threshold model developed by Bonabeau, the experiments have been implemented in TeamBots [9].

C. Multi-robot systems

One of the most popular approaches based on auction market mechanisms for the coordination of multi-robot systems was introduced by Dias and Stentz [10] in 2000. They consider that in multi-robot systems based on auctions, the robots are designed as agents of their own interests operating in a virtual economy. The tasks are assigned to the robots through the auction market mechanisms, for each task the complete robot generates some income that are reflected in the form of virtual money for providing a service to the team. However, when executing a task, the robot consumes resources such as fuel or network bandwidth, therefore, requires some expenses to pay for the resources used to complete the task. In 2004 [11] Dias has developed a coordination mechanism called *Traderbots*, which is designed to inherit the effectiveness and flexibility of a market economy. In this approach, were made some improvements in relation to the estimated costs to improve the efficiency of the team, then, in 2006 [12] this mechanism was applied in teams of harvesting to search treasure in an unknown environments.

Shiroma and Campos proposed a framework for coordination and distribution of tasks between a set of heterogeneous mobile robots called *CoMutaR* (Coalition formation based on Multi-tasking robots), allowing the robots to perform multiple tasks same time. It is based on the Contract Net Protocol to form coalitions concurrent through actions, use an auction process of a single round. They considered two specific experiments: (1) that two robots cooperate to push a box and (2) that a set of three tasks are performed by two robots [13].

Song et al. proposed a Distributed Bidirectional Auction algorithm for multi-robot systems coordination. A task is divided into n sub-tasks, a robot can only run a sub-task, the allocation of sub-tasks is decided by both the auctioneer and bidder; the auctioneer chooses the pre-winners ordering the prices of offer, while the bidders chosen all tasks that pre-won the sub-task which has the lowest price. After the first round, the sub-tasks that were not chosen by any bidder enters a second round of auction depending on the initial price auction, this process is repeated until all sub-tasks have been completed [14].

In [15] Lim et al. presented an architecture based on the auction market for the cooperation of a team of robots. On this platform, each team of robots is controlled by a respective MRS Client program and communicating through ZigBee Wireless Personal Area Network (WPAN). Each WPAN is assigned whit a different identity (ID) so the data security of communicated information be preserved. A client program that acts as a buyer is used to deliver the tasks for users in the market. Then, a server program of tasks coordination is used to compare the buyers' demand matches the supply from sellers. These programs are based on client/server architecture and are connected through Local Area Network (LAN) using Transmission Control Protocol (TCP) and Internet Protocol (IP).

3 Response Threshold Model

Insect societies are characterized by the division of labor, communication between individuals and the ability to solve complex problems [16], and these characteristics have long been a source of inspiration and subject of numerous studies, acquiring great relevance for many researchers both in the field of robotics as in biology. On the one hand, the biologists trying to prove their theories of social insects on robots, and on the other hand, researchers in the discipline of robotics seek solutions to problems that cannot be solved by a single robot.

Seeley et al. [17] have considered the following experiment to study the collective behavior in a colony of insects, focusing on the work performed by bees to get honey. Two food sources are presented to the colony at 8:00 A.M. at the same distance from the hive: source A is characterized by a sugar concentration of 1.0 mol/1 and source B by a concentration of 2.5 mol/1. Between 8:00 A.M. and noon, source A has been visited 12 times and source B, 91 times. At noon, the sources are modified: source A is now characterized by a sugar concentration of 2.5 mol/1 and source B 0.75 mol/1. Between noon and 4:00 P.M., source A has been visited 121 times and source B only 10 times. Have shown that a bee has a relatively high probability of going to a good food source and abandon a poor food source.

Based on these observations, these simple rules of behaviors allow the bees to select the best quality source; Eric Bonabeau et al. have proposed a simple mathematical model of response thresholds for the regulation of division of labor in insect societies [18]. In this model assume that each task is associated with a stimulus or set of stimuli, so that individuals can detect information on each of the different stimulus intensity, therefore, can assess the demand for a particular task when are in contact with the stimulus associated.

Let s be the intensity of a stimulus associated with a particular task; s can be a number of encounters, a chemical concentration, or any quantitative cue sensed by individuals. A response threshold θ, expressed in units of stimulus intensity, is an internal variable that determines the tendency of an individual to response to the stimulus s and perform the associated task. More precisely, θ is such that the probability of response is low for $s < \theta$ and high for $s > \theta$. This mathematical model that satisfies this requirement is given by:

$$T_{\theta ij}(s_j) = \frac{s_j^n}{s_j^n + \theta_{ij}^n}(n > 1) \tag{1}$$

where $n > 1$ determines the steepness of the threshold. Fig. 1 shows several such response curves whit $n = 2$, for different values of θ. More clearly: for $s < \theta$, the probability of engaging task performance is close to 0, and for $s > \theta$, this probability is close to 1. Then, the probability than an individual will perform a task depends on s.

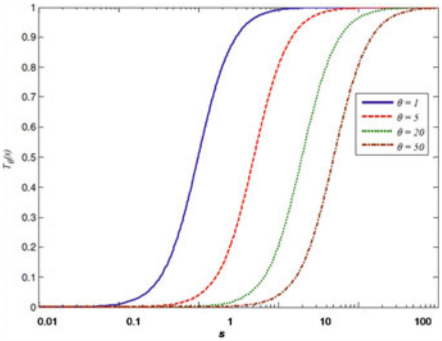


Fig. 1. Semi-logarithmic plot whit different thresholds ($\theta = 1, 5, 20, 50$) and whit $n = 2$

The underlying idea is very simple, when a stimulus exceeds the threshold of response of an individual, that individual is likely to respond to stimuli, and engage in the task because the level of the stimulus associated with that task exceeds its threshold. The intensity of a stimulus decreases as the individual performs the task; therefore, individuals with high thresholds are unlikely to perform the task when other individuals, with lower thresholds, maintain the stimulus intensity below their thresholds. However, when individuals with low thresholds do not perform the task, individuals that have high thresholds may engage in the task performance because the stimulus intensity exceeds their thresholds.

4 Formal Description of the Problem. Experimental Scenario

4.1 Formal Description of the Problem

The optimal multi-tasks allocation problem in multi-robot systems can be formally defined as follows: "Given a robot team formed by N heterogeneous robots, and given K different types of heterogeneous specialized tasks to be executed, obtain an optimal distribution of the existing tasks among the available robots". The load or number of each task can be constant (i.e, the stationary case) or can be variable and time-dependent (i.e, the dynamic case). Most of the proposed solutions in the technical literature are of a centralized nature, in the sense that an external controller is in charge of distributing the tasks among the robots by means of conventional optimization methods and based on global information about the system state [19]. However, we are mainly interested on truly decentralized solutions in which the robots themselves, autonomously and in an individual and local manner, select a particular task so that all the tasks are optimally distributed and executed. In this regard, we have experimented with bio-inspired threshold models to tackle this hard self-coordination problem as described in the sequel.

4.2 Experimental Scenario

In this area of research, we propose the following experimental scenario, with the objective of analyzing a concrete strategy or solution for the coordination of multi-robot systems as regards the optimal distribution of the existing task. Given a set of N heterogeneous mobile robots distributed within a single region or at different regions in which case they form sub-teams in each region that will perform a set of tasks T that may be divided into j sub-tasks. The sub-teams are dynamic over time, i.e. the same robots will not be always part of the same sub-team, but the components of each sub-team can vary depending on the situation.

Let $H = \{h_1, h_2, ..., h_N\}$ be the set of regions to be explored. Let T denote the task, which divided into j sub-tasks $\{t_1, t_2, ..., t_J\}$. Let $R = \{r_1, r_2, ..., r_K\}$ be the set of K heterogeneous mobile robots $(K \geq J)$. Let $S = \{s_1, s_2, ..., s_M\}$ be the set of M sub-teams by region, where $s_M \subseteq R$, i.e. the sum of $s_1 + s_2, ..., +s_M = K$.

Using the mathematical model of response threshold, to obtain the optimal solution to the problems raised here:

1. To perform the distribution of a set of K heterogeneous mobile robots in N regions for all $K = 1$ to K.
2. To perform the task allocation for each robot r_k belonging to the sub-team s_M.

To solve the problem, we make the following assumption: that each region has the same number of J tasks; that all members $R = \{r_1, r_2, ...r_K\}$ are able to participate in any sub-task $t_{j,m}$.

5 Simulation Results

In the simulations, we have conducted several experiments designed to analyze and evaluate the system performance index, we have considered some variants such as: the multi-robot system size, regions number and tasks number by region. The tasks can be constant or can be and time-dependent variable. Stimuli associated with each task can vary considerably from one task to another depending on the nature of tasks, task demand and by number of robots that are executing the task. Each task is associated with the demand expressed in the form of a stimulus, when a robot performs a task tends to reduce the intensity of associated stimulus, and as a result, modifies the intensity of the stimuli for tasks that is not running.

Each robot $\{r\}$ has a set response thresholds $\theta_r = \{\theta_1, \theta_2, ..., \theta_T\}$. Each threshold $\theta_{r,t}$ corresponds to a task type $t_j = \{t_1, t_2, ..., t_J\}$ that the robot is capable of. The initial values of the threshold are randomized to ensure that their roles are not predetermined; the performance of a given task induces a decrease in threshold of the robots:

$$\theta_{r,t}^{new} = \theta_{r,t}^{old} - \sigma \tag{2}$$

And conversely, the not performance of a given task induces:

$$\theta_{r,t}^{new} = \theta_{r,t}^{old} + \sigma \tag{3}$$

where $\sigma > 0$.

The distribution of robots in regions is done by Equation 1, obtaining a vector of probabilities of each robot for each region, can be represented as follows:

$$P(H|R = r_i) = [P(h_1|R = r_1), ..., P(h_N|R = r_1)] \tag{4}$$

where $P(H|R = r_i) \in [0, 1]$, N is the maximum region number, $i = 1, 2, ..., K$ robots.

5.1 Distribution of Robots in Regions

Fig. 2 shows the distribution of 10 robots in 4 regions. In Fig. 3 are shown the results obtained in relation with the stimuli, thresholds and probabilities for 2

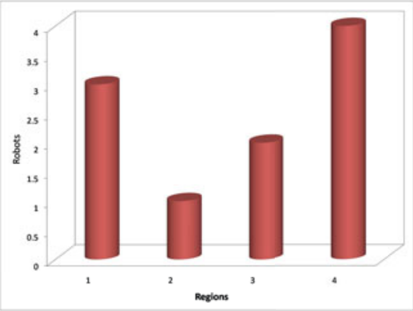


Fig. 2. Distribution of 10 robots in 4 regions

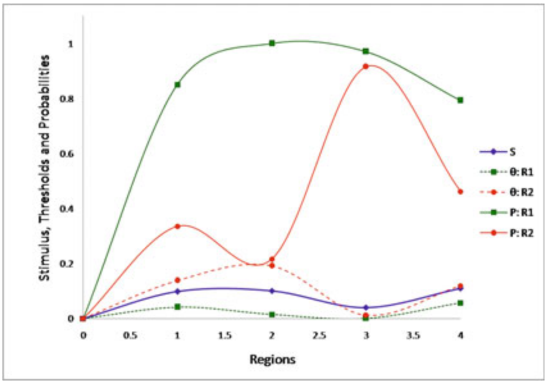
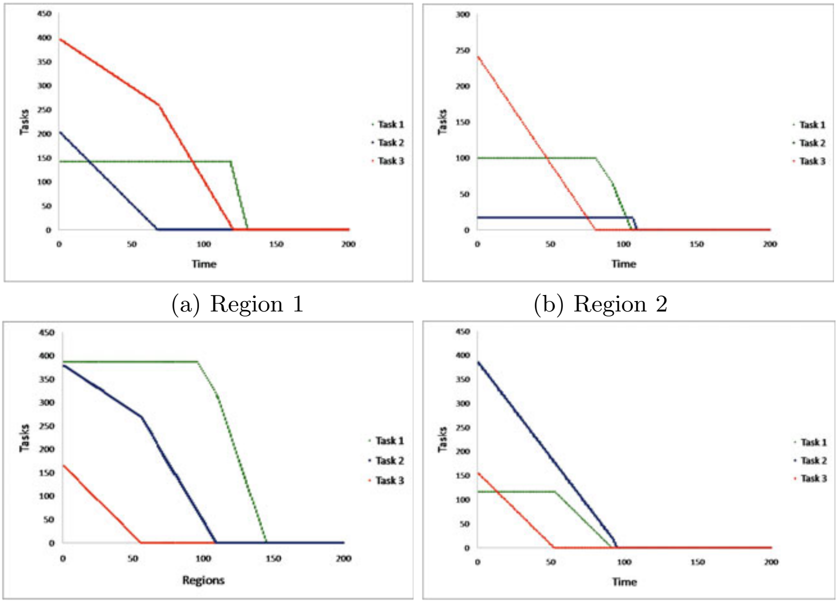


Fig. 3. The curves show the probability of response of 2 robots as a function of its thresholds and the stimulus intensity



(a) Region 1

(b) Region 2

(c) region 3

(d) Region 4

Fig. 4. Performance index of tasks by region

of the 10 robots, one can see that, when the stimulus associated with a region is above the threshold of the robot, the robot obtains a high probability to go to that region. Specifically, the stimulus of the region 2 exceeds the threshold of the robot 1, as a result has a high probability, and while for the robot 2, the stimulus is below the threshold set for the robot, therefore, the probability to go to that region is close to 0.

5.2 Task Allocation

After making the distribution of 10 robots in 4 regions, taking into account that each region has 3 types of tasks, we make the task allocation to robots using equation 1, obtaining as result the probability that the robot r_i will perform task j of type t. In Fig. 4 we present the evolution of the system performance index for each type of task and for each region.

Each time that a sub-team of robots completes all tasks associated with a region, the current probability of that region is set to 0, for not taking into account the region in the next distribution of robots.

6 Conclusions and Further Work

In this paper we have proposed an experimental scenario in which several heterogeneous and specialized tasks distributed in different regions have to be executed by a team of heterogeneous mobile robots. We have focused our interest on truly decentralized solutions in the sense that the robots have to select in an autonomous and individual manner the existing tasks so that all the tasks are optimally executed without the intervention of any global and central tasks scheduler. We have shown that the bio-inspired threshold model can be efficiently applied to solve this self-coordination problem in multi-robot systems.

We also plan to experiment in the future with alternative methods like Learning Automata-based probabilistic reinforcement learning algorithms [20] as we believe they are well tailored to solve hard on-line learning problems like the one tackled in this paper.

References

1. Parker, L.E.: Multiple Mobile Robot Systems. In: Bruno, S., Oussama, K. (eds.) Springer Handbook of Robotics, pp. 921–941. Springer, Heidelberg (2008)
2. Braunl, T.: Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems, p. 246 (2006)
3. Huntsberger, T., Trebi-Ollennu, A., Aghazarian, H., Schenker, P.: Distributed Control of Multi-robot System Engaged in Tightly Coupled Tasks. journal autonomous robots, 79–92 (2004)
4. Farinelli, A., Locchi, L., Nardi, D.: Multirobot Systems: A Classification Focused on Coordination. IEEE Transactions on Systems, Man and Cybernetics, 2015–2028 (2004)
5. Oster, G., Wilson, E.: Caste and Ecology in the Social Insects. In: Monographs in Population Biology. Princeton Univ. Press, Princeton (1978)
6. Robinson, G.: Regulation of Division of Labor in Insect Societies. Annu. Rev. Entomol., 637–665 (1992)
7. Price, R., Tiňo, P.: Evaluation of Adaptive Nature Inspired Task Allocation Against Alternate Decentralised Multiagent Strategies. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 982–990. Springer, Heidelberg (2004)

8. Baglietto, M., Cannata, C., Capezio, F., Grosso, A., Sgorbissa, A.: A Multi-robot Coordination System Based on RFID Technology. In: IEEE International Conference on Advanced Robotics, pp. 1–6 (2009)
9. Yang, Y., Zhou, C., Tian, Y.: Swarm Robots Task Allocation Based on Response Threshold Model. In: IEEE International Conference on Autonomous Robots and Agents, pp. 171–176 (2009)
10. Dias, B., Stentz, A.: A Free Market Architecture for Distributed Control of a Multirobot System. In: 6th International Conference on Intelligent Autonomous Systems, pp. 115–122 (2000)
11. Dias, B.: Traderbots: A new Paradigm for Robust and Efficient Multirobot Coordination in Dynamics Environments. Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh (2004)
12. Jones, E., Browning, B., Dias, B., Argall, B., Veloso, M., Stentz, A.: Dynamically Formed Heterogeneous Robot Teams Performing Tightly-Coordinated Tasks. In: IEEE Conference on Robotics and Automation, pp. 570–575 (2006)
13. Shiroma, P., Campos, M.: CoMutaR: A Framework for Multi-robot Coordination and Task Allocation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4817–4824 (2009)
14. Song, T., Yan, X., Liang, A., Chen, K., Guan, H.: A Distributed Bidirectional Auction Algorithm for Multirobot Coordination. In: IEEE International Conference on Research Challenges in Computer Science, pp. 145–148 (2009)
15. Lim, C., Mamat, R., Brunl, T.: Market-Based Approach for Multi-team Robot Cooperation. In: IEEE International Conference on Autonomous Robots and Agents, pp. 62–67 (2009)
16. Bonabeau, E., Theraulaz, G., Deneuborurg, J.: Quantitative Study of the Fixed Threshold Model for the Regulation of Division of Labour in Insects Societies. In: Proceedings Biological Science, pp. 1565–1569 (1996)
17. Seeley, T., Camazine, S., Sneyd, J.: Collective Decision-Making in Honey Bees: how Colonies Choose Among Nectar Sources. Behavioral Ecology and Sociobiology, 277–290 (1991)
18. Bonabeau, E., Theraulaz, G., Deneubourg, J.: Fixed Response Thresholds and the Regulation of Division of Labor in Insect Societies. Bulletin of Mathematical Biology, 753–807 (1998)
19. Gerkey, B., Mataric, M.: Multi-Robot Task Allocation: Analyzing the Complexity and Optimality of Key Architectures. IEEE International Conference on Robotics and Automation, 3862–3868 (2003)
20. Maravall, D., De Lope, J.: Fusion of Learning Automata Theory and Granular Inference Systems: ANLAGIS. Applications to Pattern Recognition and Machine Learning, Neurocomputing (2010)

An Incremental Model of Lexicon Consensus in a Population of Agents by Means of Grammatical Evolution, Reinforcement Learning and Semantic Rules

Jack Mario Mingo[1] and Ricardo Aler[2]

[1] Computer Science Department, Autonomous University of Madrid
mario.mingo@uam.es
[2] Computer Science Department, Carlos III University of Madrid
aler@inf.uc3m.es

Abstract. We present an incremental model of lexicon consensus in a population of simulated agents. The emergent lexicon is evolved with a hybrid algorithm which is based on grammatical evolution with semantic rules and reinforcement learning. The incremental model allows to add subsequently new agents and objects to the environment when a consensual language has emerged for a steady set of agents and objects. The main goal in the proposed system is to test whether the emergent lexicon can be maintained during the execution when new agents and object are added. The proposed system is completely based on grammars and the results achieved in the experiments show how building a language starting from a grammar can be a promising method in order to develop artificial languages.

Keywords: Swarm Intelligence, Grammatical Evolution, Language Acquisition and Language Development.

1 Introduction

Traditionally the study of language has been promoted by psychologists, biologists, linguists, philosophers and other researchers of human and social sciences. With the increasing power and flexibility of computers a new via in the study of the language raised. Mostly, these computational studies try to analyse the human language using simulations (see [1] or [2] for a detailed review). Fewer studies have been proposed only to develop a language for a population of robots or simulated agents. The system we present here corresponds to this approach.

The first attempts in elaborating complex communication for artificial agents is being devoted to create a basic vocabulary or lexicon that it is shared by all the agents. In the context of human communication, grammars have been used to explain the language acquisition and development process since Chomsky popularized them in the fifties [3]. Curiously, grammars are not extensively used to build artificial languages for artificial organisms. A pioneer work in this line

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 40–49, 2011.

was [4]. Other works have been presented to develop a common language or vocabulary and we will mention only a few ones as [5], [6], [7] or [8]. A good review about robotic and language evolution is reported in [9]. These methods and other ones have important advantages but we think grammars have most expressive power. Another important advantage of grammars is the chance for expressing a potentially infinitive number of structures with a finite device.

The main aim of this work is to propose an incremental model for creating a consensual lexicon in a population of artificial agents. We look for this goal by means of a hybrid algorithm which merge grammatical evolution with semantic rules and reinforcement learning. In the incremental model the system tries to evolve a lexicon for a steady set of objects and agents. When the lexicon consensus has emerged new objects and agents can be added to the environment. Then, the system continues and it tries to find a new consensual lexicon with the new objects. However, it is very important to maintain the previous knowledge, i.e. the word-meaning mappings must be preserved because if not the system would change continuously the lexicon. The experiments show how a language only with lexicon components can be developed and acquired for a group of agents and how we can add new objects and agents in order to enhance the vocabulary.

2 Basic Concepts

We need to define some preliminary concepts before to describe the algorithm and the experiments. A *vocabulary* or *lexicon* is an association between a meaning and a symbol. A vocabulary is *consensual* when all the agents share the same associations. A *meaning* can refer to an object or a situation but in this work we use meanings only as a reference for objects. A *symbol* or *word* is a different way of referring to an object. Both meanings and symbols refer objects and they define a reference but they do it in a different description level. The idea of different description levels has been proposed by Deacon [10] or Harnard [11]. In the Deacon and Harnards' proposal a meaning could be a mental representation, an icon or whatever item we can build as a consequence of the internal transformation corresponding to the projection of the object in the mind. We simplify this step in this work and we will implement it in a very simple way as we will show later. In a second description level, once the physical connection has been established, the agents can associate each meaning with a symbol. Mapping between objects and meanings is a complicated problem known as the *Symbol Problem Grounding* and we are not going to analyse it deeply in this work. Some elaborated proposals about how to manage this problem has been reported in [7], [11] or [12].

In this work we follow a behaviour-based approach (see [13] for details) in the design of the agents. A simulated agent is endowed with basic behaviours for acting and sensing. The perception mechanism allows to distinguish objects and agents. Besides the motor and perceptive behaviours we include two additional specialized behaviours. The first specialized behaviour is devoted to associate symbols with meanings and it is triggered when the agent perceives a specific object. The second specialized behaviour is supplied to play language games. *Languages games* are a very useful tool when we study the language. Some works in robotics propose language games as a model of communication (see [7], [8], [12] or [14]). In the system we propose here a *language game* is a dialogue between two agents. During the dialogue all agents communicate each other their current symbol-meaning mappings. The more coincidence in the mappings the more consensus the system will attain. When all symbol-meaning mappings are equal we can affirm that a lexicon consensus has emerged. Traditionally, in the computational implementations of language games, the dialogues are realized between all the agents in a hand-made design. However, in this case the language games depend on interactions between agents, i.e. only when two agents are close enough with each other a dialogue is produced. We think this choice is more realistic.

3 Grammatical Evolution Guided by Reinforcement and Semantic Rules

Grammatical Evolution Guided by Reinforcement (GER) is an extension of the Grammatical Evolution (GE) [15] and it was proposed initially in [16]. GER tries to merge evolution and learning in a simple way: it allows that individuals of the population can rewrite its own code several times, trying different choices with the same BNF grammar that they used for mapping the original genotype in the phenotype. The process in GER consists of three stages:

1. *Transcription* is the process that transforms the original binary string into an integer number string.
2. *Translation* uses the integer number string for getting a value that represents the rule to apply for the current non terminal in the grammar.
3. *Learning* stage uses a reinforcement learning mechanism for generating new programs. Specifically the reinforcement learning used in GER is based on Q-Learning [17].

Stages 1 and 2 were defined in GE while stage 3 was added in GER. Essentially, the learning process in GER allows to generate new programs by means of a reinforcement mechanism instead of using the evolutionary mechanism. The reinforcement is based in the Q-Learning's idea and GER implements it by means of an structure known as the Q-Tree. The Q-Tree is similar to the traditional Q-table, but it allows to work with a potentially larger space of states and actions. The Q-tree supports the production rules more suitable on each state. Nodes in the Q-tree stands for states in the process of building a derivation tree which represents a program. Q-tree keeps the next information on each node:

- A numeric value that represents the production rules used to reach the state.
- A group of numeric values called Q-values, which represent the reward associated with the different production rules that can be applied in the state.

When the learning process starts, GER uses and maintains the Q-tree in order to select a production rule in the grammar. GER tries to select the fitter production rule by using the Q-values. The Q-value represents a reward and it is computed and updated when the individual is evaluated during the evaluation process.

An important drawback in GE and GER is that they only can guarantee syntactically correct programs but none about semantic correctness is assured. With the aim to solve this drawback a GER with semantic rules is recently being developed [18]. Semantic is included via translation schemes [19]. We have no space here to describe the process with detail but semantic rules represent semantic constraints and they consist of attributes and rules. They appear in the original grammatical rules in order to avoid that semantically wrong individuals can be generated. Both GE and GER can build whatever structure we can describe by means of a grammar and we will use it to describe a vocabulary made up of words. Thus, following grammar would describe a simple vocabulary with the words *wall*, *box*, *ball* and *dice*.

```
<vocabulary> ::= <word-set> [<vocabulary>.value='vocabulary']  (0)
<word-set> ::= <word>       [<word-set>.value=<word>.value] (0)
    | <word-set> ; <word>
     [if <word-set>.value "contains" <word>.value then semantic=wrong]
     [<word-set>.value=<word-set>.value+<word>.value]       (1)
<word> ::= wall [<word>.value='wall'] (0)
    | box [<word>.value='box']        (1)
    | ball [<word>.value='ball']      (2)
    | dice [<word>.value='dice']      (3)
```

Numbers in the right side are used during the translation and learning processes in order to choose the rules. Alternatively, sentences in square brackets represent the semantic constraints and they can be interpreted as follow: each non-terminal has an attribute *value* and some non-terminal can include a semantic constraint as a semantic rule. In the example, the rule *if <word-set>.value contains <word>.value then semantic=wrong* avoids that individuals with two o more repeated words can be generated. This way, we discard individuals with repeated words because they are useless.

4 Incremental Model and Environment

4.1 Incremental Model of Lexicon Consensus

Basically, the incremental model is held by a modified GER. The modified version of GER tries to maintain the solutions previously found by an original GER. To clarify how the incremental model is implemented we summarize the algorithm in figure 1.

Fitness value is calculated from two contributions. Firstly, we consider the individual's skill in order to generate unique words in the vocabulary. The maximum value would be the maximum number of unique words in the vocabulary.

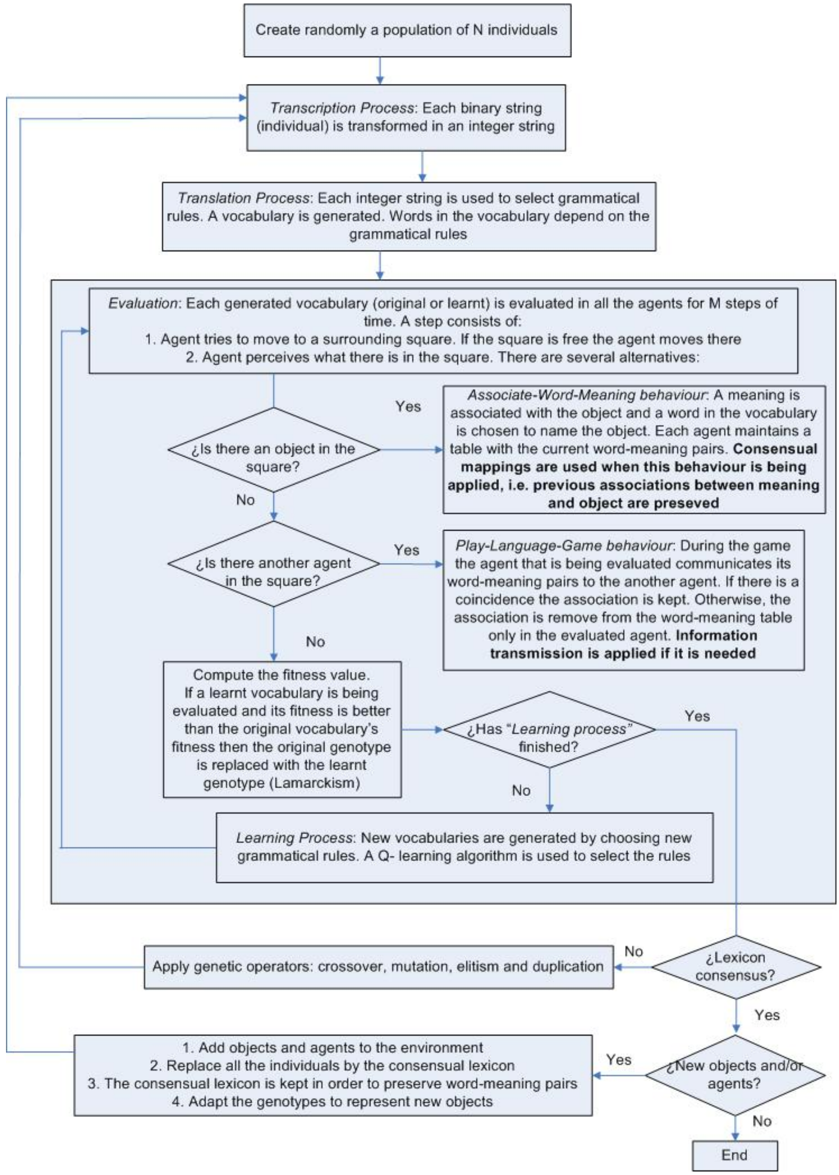

Create randomly a population of N individuals
Transcription Process: Each binary string (individual) is transformed in an integer string
Translation Process: Each integer string is used to select grammatical rules. A vocabulary is generated. Words in the vocabulary depend on the grammatical rules
Evaluation: Each generated vocabulary (original or learnt) is evaluated in all the agents for M steps of time. A step consists of:
1. Agent tries to move to a surrounding square. If the square is free the agent moves there
2. Agent perceives what there is in the square. There are several alternatives:
¿Is there an object in the square?
Yes
No
Associate-Word-Meaning behaviour: A meaning is associated with the object and a word in the vocabulary is chosen to name the object. Each agent maintains a table with the current word-meaning pairs. **Consensual mappings are used when this behaviour is being applied, i.e. previous associations between meaning and object are preseved**
¿Is there another agent in the square?
Yes
No
Play-Language-Game behaviour: During the game the agent that is being evaluated communicates its word-meaning pairs to the another agent. If there is a coincidence the association is kept. Otherwise, the association is remove from the word-meaning table only in the evaluated agent. **Information transmission is applied if it is needed**
Compute the fitness value. If a learnt vocabulary is being evaluated and its fitness is better than the original vocabulary's fitness then the original genotype is replaced with the learnt genotype (Lamarckism)
¿Has "*Learning process*" finished?
Yes
No
Learning Process: New vocabularies are generated by choosing new grammatical rules. A Q- learning algorithm is used to select the rules
Apply genetic operators: crossover, mutation, elitism and duplication
No
¿Lexicon consensus?
Yes
1. Add objects and agents to the environment
2. Replace all the individuals by the consensual lexicon
3. The consensual lexicon is kept in order to preserve word-meaning pairs
4. Adapt the genotypes to represent new objects
Yes
¿New objects and/or agents?
No
End

Fig. 1. Incremental model algorithm

Secondly, the vocabulary must be shared by all the agents. The first contribution stands for an innate factor in the individual because it comes directly from the genotype. The second contribution would imitate a learning factor because it comes from the evaluating process, i.e. it is computed when the agent is executing. Next formula shows how the fitness value is computed.

$$fitness = 1 - [WUW * (uW/mUqW) + WWMS * (wmSP/(mWMP * nA))] \quad (1)$$

where WUW and $WWMS$ stand for a percentage we assign to each contribution, uW means the number of unique words in the generated vocabulary, $mUqW$ means the maximum number of unique words that we can build with the grammar, $wmSP$ represents the number of successes in the language games, i.e. the number of right word-meaning pairs, $mWMP$ stands for the maximum number of associations and nA is the number of agents in the environment. We subtract 1 because in GER a minimum value of fitness is better.

4.2 Environment and Experiments

To elaborate the experiments we checked several robots simulators but finally we decided to build a simple model because we are only interested in analysing the language and we do not need to devote too much time with low level functions as it is usual when we work with perceptive and locomotive skills. We use a closed grid environment with a size of 15x15 squares where a maximum of 16 objects and 16 agents can be situated. Walls are considered like an object and we use them as separators. All the experiments initially start with four objects (*wall, box, ball and dice*) and four agents. Figure 2 shows an example of the environment with the initial objects randomly situated.

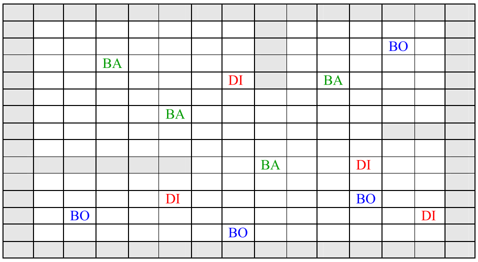


Fig. 2. Environment with walls (light gray), balls (BA), boxes (BO) and dices (DI)

Agents are also randomly located in the environment and they can move in horizontal, vertical or diagonal but they only can fill a square if it is free. Each agent has sensors to check what there is on a square. As we mentioned the *symbol grounding problem* is simplified and we associate the combinations 'WA', 'BO', 'BA' and 'DI' as representations for the meaning of the initial objects. Although we use letters for clarity whatever representation way we choose would be similar. The initial set of objects can be enhanced when GER has found a consensual lexicon. Thus, the complete set of symbols (S) and meanings (M) we work in the experiments is the following:

- $S = \{wall, box, ball, dice, feather, glass, pencil, sharpener, cherry, orange, apple, pear, strawberry, lemon, melon, nut\}$
- $M = \{'WA,'BO','BA','DI','FE','GL','PE','SH','CH','OR','AP', 'PR','ST','LE','ME',NU'\}$

When a new object or agent is added it appears in the environment and a new rule is added to the grammar in order to allow that the new symbol can be included in the vocabulary. With regard to the experiments we address a set of 5 executions. On each execution we could add two new objects and two new agents each time. The evolutionary parameters were established as table 1 shows.

Table 1. Evolutionary Parameters

Population size	20
Maximum generations	100
Learning steps	10
Crossover, Mutation and Duplication probabilities	0.9 ; 0.03 ; 0.05

5 Results

Figure 3 shows the cumulative success frequency. This value measures the probability of finding a solution to a problem in a specific number of generations.

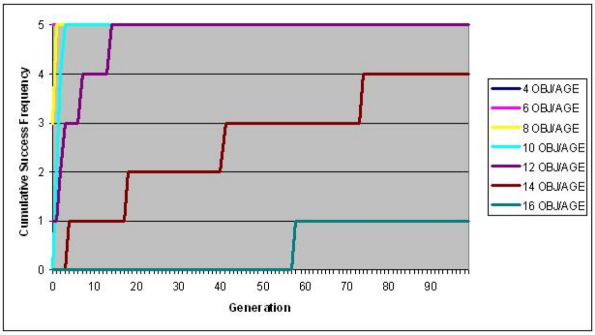


Fig. 3. Results of Cumulative Success Probability after 5 executions

In the case of 4, 6, and 8 symbols the incremental model is able to find a solution just in the first or second generation. For 10 objects a solution is always found before the fourth generation. The incremental model found a solution before the fourteenth generation when 12 objects and 12 agents were situated in the environment. In the case of 14 and 16 objects the incremental model was not able to find a solution in all the executions but it performs acceptably with 14 objects because it found a solution in four executions (success 80%). However,

Table 2. Results of an individual execution

SYMBOL	G0	G0	G0	G1	G0	G1	G0	G1	...	G3	G0	G1	...	G4	G0	G1	...	G58
dice	wa	wa	wa	wa	wa	wa	wa	wa	...	wa	wa	wa	...	wa	wa	wa	...	wa
ball	bo	bo	bo	bo	bo	bo			...	bo			...	bo	bo	bo	...	bo
box	di	di	di	di	di	di	di	di	...	di	di	di	...	di	di	di	...	di
wall	ba	ba	ba	ba	ba	ba	ba	ba	...	ba	ba	ba	...	ba	ba		...	ba
feather		gl	gl	gl	gl	gl	gl	gl	...	gl	gl	gl	...	gl		gl	...	gl
glass		fe		fe	fe	fe	fe	fe	...	fe	fe	fe	...	fe	fe	fe	...	fe
pencil			pe	pe	pe	pe	pe	pe	...	pe	pe	pe	...	pe		pe	...	pe
sharpener			sh	sh	sh	sh	sh	sh	...	sh	sh	sh	...	sh	sh	sh	...	sh
orange						ch	ch	ch	...	ch	ch	ch	...	ch	ch	ch	...	ch
cherry					ch	or	or	or	...	or	or	or	...	or			...	or
apple									...	ap	ap	ap	...	ap	ap	ap	...	ap
pear							ap	ap	...	pr	pr	pr	...	pr	pr	pr	...	pr
strawberry									...		st	st	...	le	le	le	...	le
lemmon									...				...	st	st	st	...	st
melon									...				...		me	me	...	nu
nut									...				...		nu	nu	...	me
objects	4	6	8	8	10	10	12	12	...	12	14	14	...	14	16	16	...	16
agents	4	6	8	8	10	10	12	12	...	12	14	14	...	14	16	16	...	16
fitness	0	0	0.12	0	0.1	0	0.16	0.16	...	0	0.14	0.14	...	0	0.18	0.15	...	0

the incremental model only performed well in one execution for 16 objects and 16 agents with the evolutionary parameters shown in table 1 (success 20%). Table 2 shows the results of the fifth execution, the only one when a solution for the 16 objects was found.

Information in the header row describes the titles for the symbols and the generations where a lexicon consensus emerged. Next rows show the consensual symbol-meaning pairs and we must translate the information as follow: in the intersection for the symbol *dice* (first column) and the generation *G1* (fifth column) the meaning *wa* means that all the agents in the environment agreed to associate the symbol *dice* with the meaning *wa* in the first generation. The association can be done because the generated vocabulary included the appropriated symbol. In this case the symbol *dice* must be in the vocabulary so the association can be generated when the vocabulary is being evaluated. Three last rows describe the number of objects and agents in the environment and the fitness value. A fitness value of zero is attained once the lexicon consensus has emerged. At that time, a new configuration starts. Symbols column shows how many words are in the vocabulary. For example, in the second column titled as *G0* we can see how only a meaning for the symbols *dice, ball, box* and *wall* is represented because a vocabulary with those words was being evaluated. Blank spaces in table 2 stand for symbols out of the grammar or symbols in the grammar but they did not attain an association between the symbol and a meaning. For example, if there is only four objects in the environment we only have four symbols in the grammar. It is worthy to mention how the symbol-meaning pairs are held through the configurations. In our opinion this is the most outstanding aspect in the system because it shows how the incremental model is able to maintain the previous lexicon consensus. We think this is a key factor in the language developing and acquisition process because the language probably would not evolve as quick and reliable as it really does if the consensus were not maintained through generations.

6 Conclusions

Multi-agent systems would probably improve their skills and usefulness if they were able to communicate each other with a complex communication system. First attempts in artificial communication is being focused in developing a vocabulary or lexicon. With a consensual lexicon we can address more complex and more ambitious goals about artificial language. In this work we propose an incremental model to develop a consensus lexicon in a population of simulated agents by means of a hybrid algorithm which merge grammatical evolution with semantic rules and reinforcement learning. Although the experiments we addressed in this document could be developed with other methods we use grammar because in a future we think to enhance the system in order to generate a more complicated language with a syntactical structure. In this case, we think grammars are the most efficient mechanism. Meanwhile, this work allows us to stablish an starting point.

The proposed incremental model allows to add new objects and agents to the environment when a consensus lexicon has emerged for an initial set of objects. Each object is named with a symbol and a meaning for the object is associated with the symbol by means of language games. Agents are endowed of basic behaviours for sensing and acting and two specialized behaviours for associating meaning and symbols and for playing language games. Language games are only developed when two agents find each other in order to simulate a realistic situation. Results show how the incremental model is able to find solutions for consensual vocabularies with 4, 6, 8, 10, 12 and 14 symbols and agents with a limited set of resources (a low value of individuals in the population and few generations). Thanks to grammars the solutions are more readable and understandable as well. The incremental model shows how the consensual lexicon can be maintained through configurations and how the previous symbol-meaning pairs are not lost.

Acknowledgements

This work has been funded by the Spanish Ministry of Science under contract TIN2008-06491-C04-03.

References

1. Briscoe, E.J.: Linguistic Evolution through Language Acquisition: formal and computational models. Cambridge University Press, Cambridge (2002)

2. Kirby, S.: Natural Language from Artificial Life. Artificial Life 8(3), 185–215 (2002)
3. Chomsky, N.: Three Models for the Description of Language. I.R.E Transactions on Information Theory IT-2, 113–124 (1956)
4. Hashimoto, T., Ikegami, T.: Evolution of Symbolic Grammar Systems. In: Morán, F., Merelo, J.J., Moreno, A., Chacon, P. (eds.) ECAL 1995. LNCS, vol. 929, pp. 812–823. Springer, Heidelberg (1995)
5. Werner, G., Dyer, M.: Evolution of Communication in Artificial Organisms. In: Langton, C., et al. (eds.) Artificial Life II, pp. 659–687. Addison-Wesley Pub. Co., Redwood City (1991)
6. Yanko, H., Stein, L.: An Adaptive Communication Protocol for Cooperating Mobile Robots. In: Meyer, J.-A., Roitblat, H.-L., Wilson, S. (eds.) Proceedings of the Second Interntational Conference on Simulation of Adaptive Behavior,From Animals to Animats 2, pp. 478–485. The MIT Press, Cambridge (1993)
7. Steels, L.: The Spontaneous Self-organization of an Adaptive Language. In: Muggleton, S. (ed.) Machine Intelligence 15. Oxford University Press, Oxford (1996)
8. Maravall, D., de Lope, J., Domínguez, R.: Self-emergence of Lexicon Consensus in a Population of Autonomous Agents by Means of Evolutionary Strategies. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS, vol. 6077, pp. 77–84. Springer, Heidelberg (2010)
9. Vogt, P.: Language Evolution and robotics: Issues on symbol grounding and language acquisition. In: Loula, A., Gudwin, R., Queiroz, J. (eds.) Artificial Cognitive Systems, pp. 176–209. Idea Group, Herschey (2006)
10. Deacon, T.: The symbolic species: The co-evolution of language and the human brain. Penguin Books (1997)
11. Harnard, S.: The symbol grounding problem. Physica D 42, 335–346 (1990)
12. Vogt, P.: Lexicon Grounding on Mobile Robot. Ph. D. Thesis. Vrije Universiteit Brussel (2000)
13. Arkin, R.: Behaviour-Based Robotic. MIT Press, Cambridge (1998)
14. Looveren, J.V.: Robotic experiments on the emergence of a lexicon. In: Ben, et al. (eds.) Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence, The Netherlands, Amsterdam (2001)
15. Ryan, C., Collins, J.J., Neill, M.O.: Grammatical Evolution: Evolving Programs for an Arbitrary Language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP 1998. LNCS, vol. 1391, pp. 83–95. Springer, Heidelberg (1998)
16. Mingo, J.M., Aler, R.: Grammatical Evolution Guided by Reinforcement. IEEE Congress on Evolutionary Computation, 1475–1482 (2007)
17. Watkins, C., Dayan, P.: Q-Learning. Machine Learning 8, 279–292 (1992)
18. Mingo, J.M., Aler, R., Maravall, D.: Static and Dynamic Multi-Robot Coverage with Grammatical Evolution guided by Reinforcement Learning and Semantic Rules (Not published)
19. Aho, A.V., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques and Tools. Addison-Wesley, Reading (1986)

Towards an Evolutionary Design of Modular Robots for Industry

Andrés Faíña, Francisco Bellas, Daniel Souto, and Richard J. Duro

Integrated Group for Engineering Research,
Universidade da Coruña, 15403, Ferrol, Spain
{afaina,fran,dsouto,richard}@udc.es
http://www.gii.udc.es

Abstract. We are interested in the next generation of industrial robots, those that are able to operate in dynamic and unstructured environments and, consequently, that are able to adapt to changing circumstances or to work on different tasks in an autonomous way. In this sense, multirobot systems and, in particular, modular systems present several features like scalability, fault tolerance, low maintenance or reconfiguration capabilities that make them highly suitable for this kind of environments. The work presented here is concerned with the problem of automatically obtaining the morphology and control structure for this type of modular systems. In this line, we present the first results produced using a newly designed constructive evolutionary approach that takes into account the extreme difficulty of the tremendously deceptive and uninformative search space this type of applications are faced with. As an example, the algorithm is used to design the morphology and the distributed control parameters for a typical benchmark problem, that of moving as far as possible in a straight line, for a heterogeneous modular robotic system developed by our group.

Keywords: Multi-robot Systems, Modular Robots, Evolutionary Algorithms, Industrial Automation.

1 Introduction

The typical production lines found in industry are highly automated using dedicated robots that perform their tasks in a very accurate and fast manner. These environments are static and structured. Consequently, the robots are always in a controlled position and their sensors and actuators can be adjusted assuming predictable conditions. In order to extend the advantages of robotic automation to other types of industries, new robots are required that can deal with dynamic and unstructured environments, such as those found in shipyards or civil engineering projects and that can easily reconfigure to perform different tasks. Some, albeit few, robots have been designed and tested for use in these types of environments. The first example that can be found in civil engineering is the SSR-1, which was used for the application of a fire protection layer to beams. Additional

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 50–59, 2011.

examples can be found in the still timid robotic presence within the shipbuilding sector in tasks like hull water-blasting [1] or welding [2]. However, automation of these types of dynamic and unstructured industrial environments is in an early stage, and, although robots are becoming more autonomous and flexible, in practice they are still designed for single tasks, with improved adaptive capabilities, but with no capacity at all in terms of reconfiguration.

It is with the objective of adding more flexibility to the systems and allowing them to be able to adapt to different tasks in different ways that multicomponent robotic systems (MCR) come into play. Multicomponent robotic systems present several features like scalability, fault tolerance, low cost of manufacturing and maintenance or reconfiguration simplicity that make them highly suitable for the kind of environments considered here. According to [3] MCR systems can be classified into three main types depending on the degree of coupling among robots: Uncoupled Distributed Systems, Modular Systems and Linked System. Here we are interested the second type, modular systems. These systems are based on the aggregation of modules with limited individual capabilities that, through rigid physical couplings between components, can produce robotic units with different physical and functional properties.

Developing a modular MCR system implies solving two problems: the mechanical and electrical design of the individual modules and the development of a procedure for their aggregation in order to obtain the final morphology and control structure for a particular task or set of tasks. There are several examples in the literature of modular robot architectures such as polybot [4], M-TRAN [5] or CONRO [6]. Most of these architectures have been designed as homogeneous sets of modules [12][5][8] and with a scientific purpose of exploring the domain and obtaining some basic principles, seldom taking into account the real constraints of industrial operation. An exception in this line is the work of Shen et al. [8] [7] with the Superbot system. Here, the modules that make up the robotic unit are an improved version of those in the M-TRAN and CONRO systems. They have been designed for unsupervised real environment operation, resisting abrasion and physical impacts, and including enhanced sensing and communications capabilities. This modular system is capable of climbing a rope or going up stairs.

In terms of obtaining the morphology, deciding on what modules and how they are connected, and the control system for the modular robot to perform a given task or set of tasks, this has usually been done by hand. Obviously this limits the applicability of the robots in reality as all their configurations would have to be pre-designed and, in most cases, implemented by a human operator. The reason for is this the surprisingly high dimensionality and complexity of the search spaces that are induced by the possible robot morphologies as well as how deceptive these spaces are, making most automatic approaches basically perform as a random search. Several authors have applied evolutionary techniques to solve in very simple instances this automatic design problem with successful results [9] [10]. But the main problem we have found in these approaches is that the evolutionary algorithm that is typically applied has not been adapted for the particularities of the modular systems and thus their scalability is limited. Basically, most authors focus their attention in the genotypical representation of the modular system, which is a complex problem, but without paying much attention to the algorithm itself.

In this paper we propose a constructive evolutionary algorithm with operators that are designed to control and adapt their exploration capabilities to this particular type of search spaces and that produce quite good results within this domain. The algorithm is applied over a heterogeneous modular architecture developed by our group in order to solve the typical benchmark problem of obtaining a modular robot that maximizes its motion in a straight line for a given amount of time. The rest of the paper is structured as follows: in section 2 we provide an overview of the heterogeneous modular system we are considering. Section 3 is devoted to the presentation of the constructive evolutionary algorithm developed in order to obtain robot morphologies and control mechanisms. An example of the application of the evolutionary algorithm to the set of selected modules in a simple task is discussed in section 4. Finally, in section 5 we will present the main conclusions of this work.

2 Heterogeneous Modular System

Most existing modular systems are made up of homogeneous sets of modules, which facilitates control and reconfiguration but limits the range of possible configurations. However, for the sake of completeness, flexibility and simplicity, the system considered here is heterogeneous and is made up of 4 types of modules:

- Actuators: those that generate robot motion, through pneumatic or electrical motors.
- Effectors: coupled to the actuator module they provide new functionalities, like legs, wheels or tools.
- Sensors: they provide external or internal information, like cameras, battery meters, etc.
- Linkers: they join other modules.

Fig. 1 displays the final design and implementation of some of the modules in this architecture. From top to bottom: slider, telescopic and rotational (axis) modules. The modules can be joined together through coupling connections in all the faces of the cubes at their extremes except one as well as through the slider in the slider module. This makes, for example, a total of 14 possible attachment positions for a slider module.

Communications are mandatory in modular robotics to assure the adequate coordination between modules. In this system, there are two types of communications paths: a CAN bus for global coordination and asynchronous local communications to allow intermodule identification (morphological propioception), and ZIGBEE wireless communications for global coordination in the case we have isolated robotic units or when the CAN bus is saturated. Each module has its own embedded microcontroller (pic32mx575f512) and both, centralized and

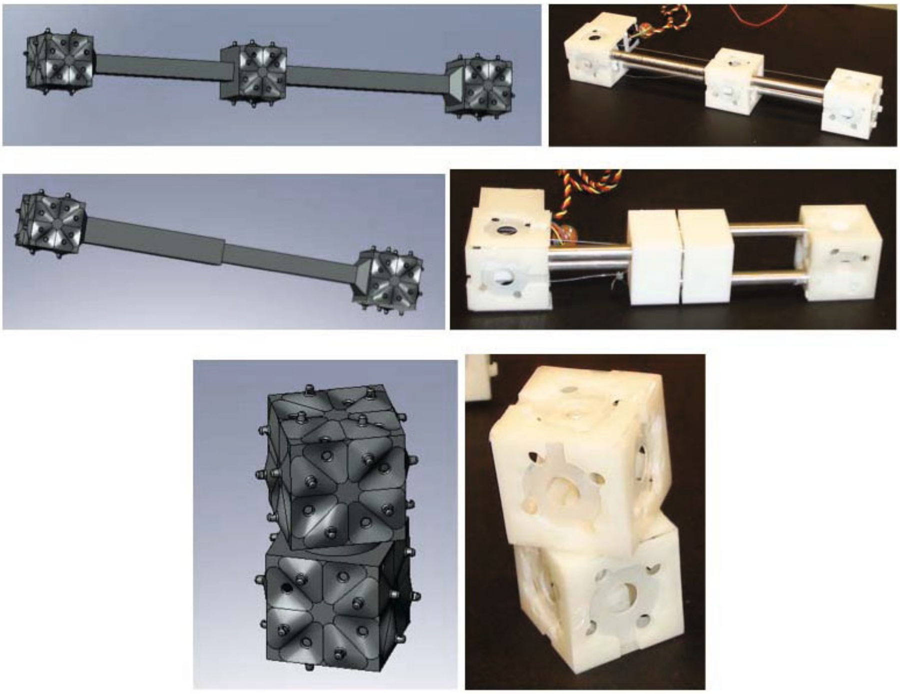


Fig. 1. Three of the modules of the architecture. Design (left) and prototype (right)

distributed control is possible. For the sake of simplicity, in this work we have chosen to use a distributed control scheme where each module can control the parameters of a sinusoidal that determines the motion of its actuator.

3 Evolutionary Design System

As indicated in the introduction, the problem we face here is that of developing an automatic design system that provides an appropriate morphology for a task or tasks together with an adequate control system. The key point in modular systems is the desired degree of autonomy. At the higher level, the optimum one, we have auto-reconfigurable adaptive systems, in which the control system has some kind of learning algorithm that allows the modular system to achieve new configurations on-line, or at least to adapt the current one to the changing circumstances. In addition, the control system must be capable of detecting the necessity of changing the task, and consequently, the configuration. This is a highly complex problem that is currently in an early stage of development in the field. As a first step to achieve such level of autonomy, here we are going to address the problem of obtaining the control system and morphology for each particular task in an off-line stage but through a general procedure, as commented above.

As the search space determined by the possible morphological configurations and control systems is huge, not necessarily continuous and usually very badly configured, we have opted for using evolution in order to obtain the robot configuration and control. However, most traditional evolutionary algorithms are not very well suited to this type of search spaces due to their intrinsic deceptiveness. This is, for instance, when trying to produce a robot that moves in a given direction, partial solutions that only lack a single module or solutions with perfect morphology but some problem in the control may not move at all or move in the opposite direction. Additionally, the evolutionary temporal scales of the morphology and the control system must be quite different even though they must be cooperatively coevolved. Finally, it is a problem where the phenotype and, consequently, the genotype is of variable size and there are all kinds of dependences between phenotypic components.

As a first approach to address these problems, we propose a constructive evolutionary algorithm. In fact, we have developed a constructive evolutionary strategy, the operation of which can be summarized into the following steps:

1. Creation of a random population of size N modular robots encoding each modular system as a tree. Each module is represented as a node with connections to child nodes. Connections contain the information about the side of the parent where the child is connected and the orientation of the child. The root node is a rectangular module that acts as a structural base with 12 connection points.
2. Mutation: we have developed a set of mutation operators that can be applied to each individual:

   – *Add Node*: this operator generates n variations of the current morphology by adding a new random module in a random position and orientation to the robot. This phase evaluates the fitness of the n variations and selects the best one replacing the parent morphology even if it is worse. This decision was taken because adding a node can cause a fitness fall in a given generation but act as a root for a higher growth in the next.
   – *Shake Modules*: this operator generates n variations of the current robot by changing the connections between modules. The change affects the side of the parent module to which the child module is attached and the orientation of the child. The fitness is calculated and the best variation is selected.
   – *Shake Control*: this operator generates n variations of the current robot by changing, with probability p, all the control parameters of the modules. The fitness is calculated for each case and the best one is selected.
   – *Delete Nodes*: this operator produces all the possible pruning variations of a robot by deleting all the modules that are downstream from a given connection. All the connections are pruned once and it is not allowed to prune two connections at the same time. So, if a robot has m modules, this operator will produce $m-1$ robots. The fitness is calculated for each case and the best one is selected.

3. Replacement: 10% of the individuals are eliminated from the population and replaced by variations of the best 5% robots and by variations of random robots. These variations are obtained through a symmetry mutation operator. This operator selects a branch of a robot chromosome and replicates it in order to achieve a symmetrical branch. Rotational and reflective symmetry are currently implemented. If it is impossible to apply the symmetry operator a random robot is generated.
4. Steps 2 and 3 are repeated until a stop criterion is met.

The mutation operators defined above perform a growth phase (add node), a morphology adaptation phase (shake modules), a control adaptation phase (shake control) and a pruning phase (delete nodes), being the order of these phase the backbone of the proposed constructive approach. The default configuration implies applying each operator once per cycle, although the best results were obtained applying the three structural operators in a higher degree than the shake control operator, reinforcing the idea that morphology and control must evolve in different time scales. The algorithm does not include a crossover operator because we have found that, with the basic tree representation of the modular systems we are using, crossing two individuals is basically a random search procedure.

Regarding the calculation of fitness, it is carried out during the mutation stage as each operator is applied. Thus, when step 3 is reached, all the individuals have already been evaluated. Fig. 2 displays a diagram of the evolutionary framework used to evolve and evaluate the modular robots. It is made up of the evolutionary strategy detailed above, which is included in the Java Evolutionary Algorithm Framework (JEAF) [11] developed within our group and accessible

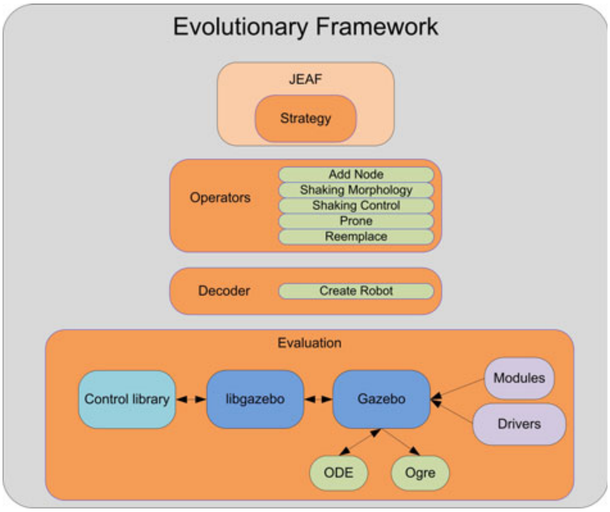


Fig. 2. Evolutionary design system

to the scientific community. JEAF contains a collection of libraries that provide all the basic functionalities of the evolutionary algorithm, represented in the figure with the block Operators. Outside JEAF we have developed a decoding library and a simulation library. For the purpose of obtaining the fitness value for the different configurations, the modular robots are simulated using Gazebo due to its real dynamics coverage. Detailed realistic models of the four types of modules established in the previous sections have been produced for these simulation. The decoding library is in charge of transforming the tree representation of an individual to a robotic unit in Gazebo, where it is run for a fixed number of steps, returning the selected fitness measure to the algorithm. In this version, the control is completely distributed over the modules, and the tree representation of each individual contains the control parameters of each module that make the robot act in the simulator.

4 Experiments

The evolutionary design system has been tested in a classical problem for modular systems: obtaining a robot able to cover the maximum distance across a flat surface [9] [10]. To this end, we have used the following actuator modules: telescopic, slider, rotational (axis) and rotational (hinge). In addition, we use a rectangular base module as an initial structure.

In this experiment, to reduce the dimensionality of the search space, the control adjustment consists in obtaining the optimal offset of the sinusoidal signal that controls each module, being the frequency and amplitude fixed. The robots are placed in the Gazebo simulated environment with, at least, one module in contact with the ground and each morphology is simulated during four cycles of the sinusoidal control signals. The fitness is calculated as the distance travelled in the X direction minus the absolute distance travelled in the Y direction. This way, we are rewarding the robotic unit that covers the maximum distance in a straight line without lateral displacement. It must be remembered that we are using a real dynamics simulation, and, consequently, unstable morphologies will obtain poor results. Related with this, to avoid the effects of robot instability in the first cycle when it is placed on the ground, we have excluded the first movement from the fitness calculation.

As commented in the previous section, the modules were designed in Gazebo in order to have realistic parameters and to be physically reproducible. The weight of the base, slider and telescopic modules is 400 g and their actuators have a force of 20 N, while the weight of the rotational modules is 200 g and their actuators' torque is 2 Nm. All of the modules have a friction coefficient of 0.8. The base has 12 connection ports: 4 on the top side corner, 4 on the bottom side corner, two on one end of the right side (parallel to the X axis) and two more on the opposite side. The slider module has two connection bays on its ends with five connection ports each and four connection ports on the mobile part. The telescopic and the rotational modules have two bays in their ends with five connections each. Finally, the hinge module has two connectors.

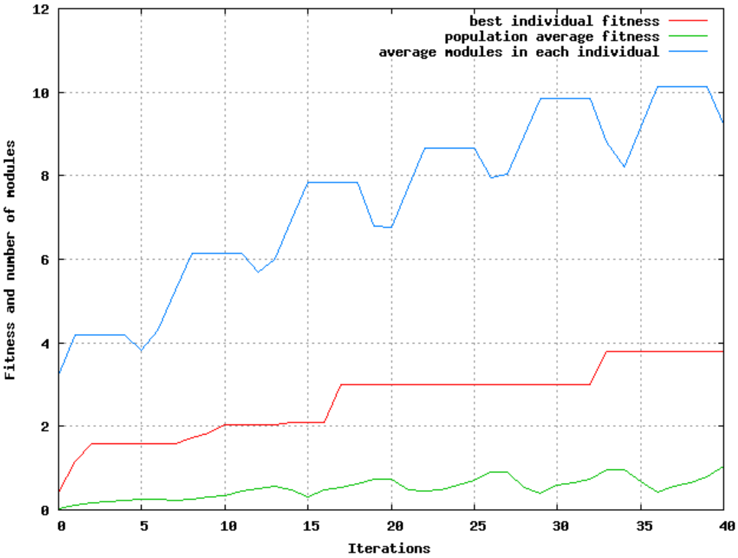


Fig. 3. Graph of a typical evolution

The evolution starts with a random population of 40 individuals and a maximum of three modules each, being 16 the maximum allowed. In this experiment, we have applied two growth phases (add node operator), two morphological mutations (shake modules operator), one control mutation (shake control operator) and two prune phases (delete nodes operator) each cycle. The growth, morphological and control phase generate three robot variations in each run. The remaining steps of the constructive evolutionary strategy are exactly as described in section 3. The algorithm was run for 10000 evaluations.

A typical evolution result is presented in Fig. 3. This graph shows the best robot fitness in red and the average fitness of the population in green. In addition, the average number of modules per robot is displayed in blue. The distinct phases of the constructive algorithm induce a cyclical pattern in the evolution as shown in the figure. The growth phase adds new modules to the robots in the population during the initial iterations. Consequently, the number of modules per robot is increased and the population average fitness is reduced because the robot is not adapted to the new module. This effect can be observed in Fig. 3 if one looks at the areas where the average number of modules line goes up, when the number of modules increase. In these zones, the average fitness decreases. Then, the morphological and control mutation phases slowly raise the average fitness and maintain the number of modules per robot. It can be observed looking at the flat zones in the average number of modules line. Finally, the pruning phase quickly raises the fitness of the population as the robots eliminate useless modules. This effect can be observed in the drop zones of the average number of modules line. The algorithm uses elitism, and that is the reason why the best individual fitness graph never oscillates.

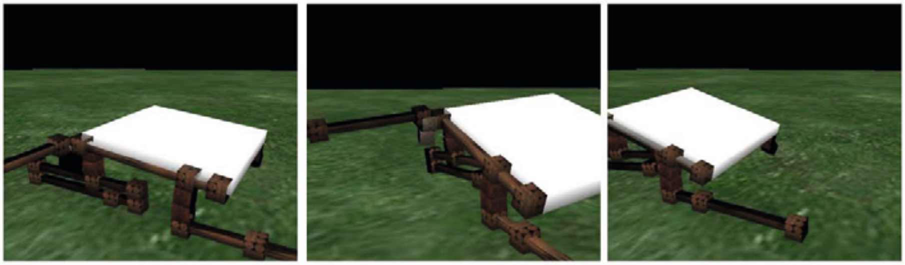


Fig. 4. Screenshots of the final morphology in action

In addition, a clear general trend can be observed in the evolution, where the number of modules per robot increases quickly during the first 30 iterations, when the algorithm is exploring the search space. In the last part of the evolution, the number of modules is stabilized to an average number of around 9 modules per robot. In practical terms, this evolution leads to a final morphology that is shown in Fig. 4 and that is able to cover a distance of around 4 m (performing a random search in the same conditions we obtain a maximum distance of 2 m). It can be observed that the robotic unit has four branches of modules that are fixed to the base. Two of them produce the movement of the robot and the other two stabilize it. Furthermore, it presents a hinge attached between the mobile part of the slider, fixed to the base, and a slider that acts like a leg. It constitutes the first branch which produces movement. The other branch is composed by a hinge fixed to the base and to the mobile part of one slider. Another slider is joined in parallel with this slider to elevate the front of the robot. In order to stabilize the robot in its motion, a slider and a rotational (axis) module are attached in the left side. The end of this slider touches the ground in a certain position of the motion and prevents the robot from falling. Finally, there is a branch made up of two hinges which are fixed at a port in the right side of the base, and which prevent the displacement of the robot in the Y axis direction.

5 Conclusions

In this paper we have introduced a constructive evolutionary strategy that permits obtaining both the morphology and the control parameters for modular multicomponent robotic systems. This algorithm has been designed to take into account the intrinsic deceptiveness and lack of information in many areas of the search space and provides an easy way to balance the different evolutionary scales at which the evolution of morphology and control must take place in order to be able to cooperatively coevolve these two elements. The algorithm has been linked to the Gazebo simulator for the evaluation of the fitness of the individuals it produces, thus obtaining a complete evolutionary design and testing system for modular robots. The operation of this approach has been shown through its application to the generation of a robot and control strategy to a typical benchmark problem using a modular architecture developed in our group with very successful results. Obviously, this is not the end of the research path, but rather its beginning. Now we have to extend this initial structure in order to consider more complex tasks and control strategies as well as start working on ways to perform the morphological and control adaptation on-line. This will involve not only deciding on the final morphology and control structure, but also on the way to get to it from the current one.

Acknowledgments. This work was partially funded by the Xunta de Galicia and European Regional Development Funds through projects 09DPI012166PR and 10DPI005CT.

References

1. Ortiz, F., Pastor, J., Alvarez, B., Iborra, A., Ortega, N., Rodriguez, D., Concsa, C.: Robots for hull ship cleaning. In: IEEE International Symposium on Industrial Electronics, pp. 2077–2082 (2007)
2. Lee, D., Lee, S., Ku, N., Lim, C., Lee, K., Kim, T., Kim, J.: Development and Application of a Novel Rail Runner Mechanism for Double Hull Structures of Ships. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3985–3991 (2008)
3. Duro, R., Graña, M., de Lope, J.: On the potential contributions of hybrid intelligent approaches to Multicomponent Robotic System development. Information Sciences 180, 2635–2648 (2010)
4. Yim, M., Duff, D., Roufas, K.: PolyBot: a modular reconfigurable robot. In: IEEE International Conference on Robotics and Automation, pp. 514–520 (2000)
5. Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., Kokaji, S.: M-TRAN: Self-reconfigurable modular robotic system. IEEE/ASME transactions on mechatronics 7, 431–441 (2002)
6. Castano, A., Behar, A., Will, P.M.: The Conro modules for reconfigurable robots. IEEE/ASME Transactions on Mechatronics 7, 403–409 (2002)
7. Ranasinghe, N., Everist, J., Shen, W.M.: Modular Robot Climbers. In: Proc. 2007 IEEE/RSJ Intl. Conf. Intelligent Robots Systems (2007)
8. Salemi, B., Moll, M., Shen, W.: SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system. In: Proc. 2006 IEEE/RSJ Intl. Conf. Intelligent Robots Systems, pp. 3636–3641 (2006)
9. Macinnes, I., Di Paolo, E.: Crawling out of the simulation: Evolving real robot morphologies using cheap, reusable modules. In: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Artificial Life, pp. 94–99 (2004)
10. Farritor, S., Dubowsky, S., Rutman, N.: On the design of rapidly deployable field robotic systems. In: ASME Design Engineering Technical Conference (1996)
11. Caamano, P., Tedin, R., Paz-Lopez, A., Becerra, J.A.: JEAF: A Java Evolutionary Algorithm Framework. IEEE Congress on Evolutionary Computation, 1–8 (2010)
12. Brandt, D., Christensen, D.J., Lund, H.H.: ATRON Robots: Versatility from Self-Reconfigurable Modules. In: Conference on Mechatronics and Automation, pp. 26–32 (2007)

Grammar-Guided Evolutionary Construction of Bayesian Networks

José M. Font, Daniel Manrique, and Eduardo Pascua

Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid. Campus de Montegancedo, 28660 Boadilla del Monte, Spain
jm.font@upm.es, dmanrique@fi.upm.es, e.pascua@alumnos.upm.es

Abstract. This paper proposes the EvoBANE system. EvoBANE automatically generates Bayesian networks for solving special-purpose problems. EvoBANE evolves a population of individuals that codify Bayesian networks until it finds near optimal individual that solves a given classification problem. EvoBANE has the flexibility to modify the constraints that condition the solution search space, self-adapting to the specifications of the problem to be solved. The system extends the GGEAS architecture. GGEAS is a general-purpose grammar-guided evolutionary automatic system, whose modular structure favors its application to the automatic construction of intelligent systems. EvoBANE has been applied to two classification benchmark datasets belonging to different application domains, and statistically compared with a genetic algorithm performing the same tasks. Results show that the proposed system performed better, as it manages different complexity constraints in order to find the simplest solution that best solves every problem.

Keywords: Evolutionary computation, Bayesian network, grammar-guided genetic programming.

1 Introduction

Bayesian networks (BN) are computational tools that can perform probabistic inference from data with uncertainty [1]. They have been applied as an automatic reasoning mechanism to a wide range of domains [2,3]. A BN is a directed acyclic graph (DAG) that codifies the existing dependencies between its nodes, each of what contains a conditional probability table [4]. The automatic learning of a BN from data is a two-step procedure composed of the design of the network topology and the calculation of its conditional probability tables [5]. This has been proven to be an NP-Hard procedure [6]. For this reason, knowledge engineering techniques need to be used to achieve quality solutions [7].

Evolutionary computation has been successfully applied to solve search and optimization problems, such as the generation of both symbolic and sub-symbolic self-adapting intelligent systems [8]. Its application to the automatic construction of BN must overcome several difficulties, such as the design of an accurate codification system that can manage acyclic graphs and the implementation of

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 60–69, 2011.

genetic operators that prevent the generation of illegal graph structures. Insofar as genetic algorithms do not solve the closure problem [9], crossover and mutation operators may generate invalid individuals. For this reason, a repair operator must be executed in order to transform invalid structures into DAGs [10].

Different evolutionary approaches codify BNs into an array of nodes that can only be forward connected [11,12,13]. In order to preserve DAG properties, crossover and mutation operators perform order permutations over this array only. The system presented in [14] replaces the crossover operator with several specific mutation operators that perform controlled mutations on the genome of the individuals. The work described in [15] implements a restricted crossover operator that reduces the search space but always produces valid offspring. In the same way, the system in [16] reduces the search space by only allowing the generation of naive-Bayes classifiers, whose straightforward codification decreases the complexity of the evolutionary procedure and avoids the need for preserving DAG structures.

This paper presents the EvoBANE (Evolutionary Bayesian Networks) system, a grammar-guided evolutionary system for automatically generating Bayesian networks that solve classification problems. EvoBANE implements a context-free grammar (CFG) generator and a fitness calculator module. The CFG generator inputs the specifications of the application domain, as well as the features of the solutions to be built. It then outputs the CFG that generates the language codifying the solution space of all the valid BN structures (individuals) for those specifications and features. The fitness calculator first calculates the conditional probability tables of the individuals by means of a probabilistic estimator [17], and then evaluates the individual's accuracy as an instance classifier. EvoBANE is based on GGEAS technology [18] to initialize and evolve a population of BN structures. Evolution is achieved by means of a general-purpose grammatical crossover operator [19] that avoids the closure problem without including a repair operator. This way, EvoBANE always generates valid individuals, preserving the DAG properties without using constraints that prevent it from exploring the whole solution space. The efficiency of EvoBANE for building Bayesian classifiers has been tested in two different application domains. A genetic algorithm with a single-point crossover operator was run on the same problems, and the results were compared.

2 The EvoBANE System

EvoBANE's structure is an extension of the modular design implemented in GGEAS. It consists of two independent components: a grammar-guided genetic programming (GGGP) core and an external layer. The GGGP core is common to every GGEAS implementation and remains unchanged whatever the chosen application domain. The external layer is composed, in this case, of two special-purpose modules: the CFG generator and the fitness calculator, whose implementation directly depends on the GGEAS application domain. EvoBANE

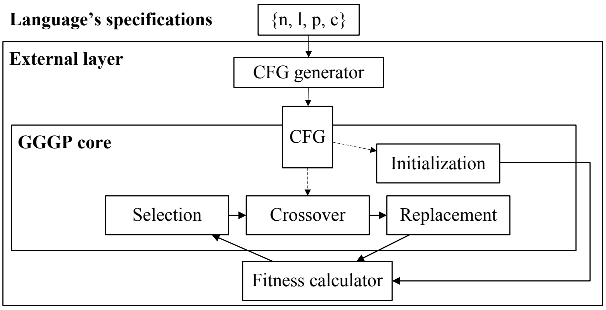


Fig. 1. Components and execution flow of EvoBANE

adopts the GGGP core and builds its own external layer modules in order to compose a system that automatically generates Bayesian networks that solve classification problems.

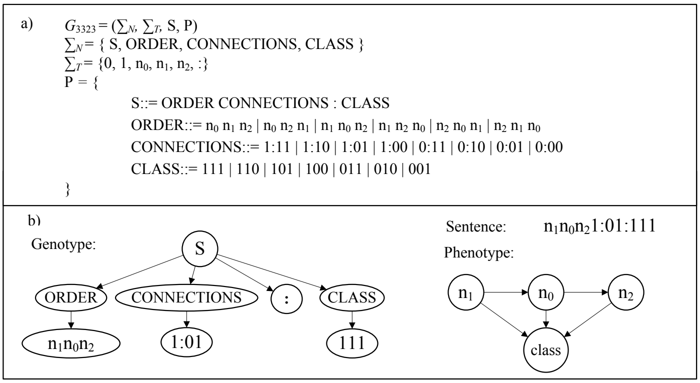
Figure 1 displays the execution flow of EvoBANE over the different components of its architecture. The CFG generator receives the specifications of the language that defines the solution space and outputs the context-free grammar that generates that language. This grammar is used by the GGGP core to initialize a population of BN structures. They are first evaluated by the fitness calculator module and then reinserted into the GGGP core for selection, crossover and replacement. The following sections explain both external modules in detail.

2.1 The CFG Generator

A context-free grammar is defined as a 4-tuple $G = (\Sigma_N, \Sigma_T, S, P) / \Sigma_N \cap \Sigma_T = \emptyset$, where Σ_N is the alphabet of non-terminal symbols, Σ_T is the alphabet of terminal symbols, S is the axiom and P is the set of production rules written in Backus-Naur form. The CFG generator in EvoBANE automatically builds grammars of this kind whose production rules generate individuals that codify valid BN structures. For this purpose, the CFG generator needs four input parameters: n, the number of nodes of the networks to be codified; l, the number of layers in which those nodes are distributed*;* p, the maximum permitted number of input connections per node and c, the maximum permitted number of input connections for the class node.

The number of nodes $n \in \mathbb{N}$ is extracted from the application domain and equals the number of feature variables observed for classifying an instance set. The class node that contains the possible classes in which instances are classified is counted as an extra node. Consequently, the total number of nodes is $n + 1$. Nodes are distributed throughout a list of l layers, where $l \in \mathbb{N}$, $1 \leq l \leq n$, and the class node alone is stored in the last layer. A layer is an array that contains a set of nodes arranged in a fixed order. Layers can be reordered to generate different BN structures, but the position of the nodes within a layer must be unchanged. Nodes arranged in layers are forward connected to each other (even within the same layer). No backward connections are allowed in order to preserve DAG properties. Connections between nodes represent their conditional dependencies. Every node has a limited number of input connections p, where $p \in \mathbb{N}$, $0 \leq p \leq n-1$, except for the class node whose limit equals c, so that $c \in \mathbb{N}$, $1 \leq c \leq n$.

Grammars generated by the CFG generator are named after the four parameters n, l, p and c in the form G_{nlpc}. These four parameters vest the CFG generator of EvoBANE with the flexibility to modify the search space of a problem depending on its complexity (given by the value of n), allowing the evolutionary system to find the simplest BN structure that best solves that problem. Insofar as the value of n is fixed, the lower the value of l, p and c is, the smaller the combinatorial explosion generated from reordering and connecting the nodes will be. Notice that a grammar created with $l = n$, $p = n - 1$ and $c = n$, generates the language of all the valid BN structures for a given n. On the other hand, a grammar with $l = n$, $p = 0$ (complete conditional independence), and $c = n$ generates the language of all the valid naive-Bayes structures for a given n.



a) $G_{3323} = (\sum_N, \sum_T, \mathrm{S}, \mathrm{P})$

$\sum_N = \{$ S, ORDER, CONNECTIONS, CLASS $\}$

$\sum_T = \{0, 1, n_0, n_1, n_2, :\}$

P = {

S::= ORDER CONNECTIONS : CLASS

ORDER::= $n_0\ n_1\ n_2 \mid n_0\ n_2\ n_1 \mid n_1\ n_0\ n_2 \mid n_1\ n_2\ n_0 \mid n_2\ n_0\ n_1 \mid n_2\ n_1\ n_0$

CONNECTIONS::= 1:11 | 1:10 | 1:01 | 1:00 | 0:11 | 0:10 | 0:01 | 0:00

CLASS::= 111 | 110 | 101 | 100 | 011 | 010 | 001

}

b)

Sentence: $n_1 n_0 n_2$1:01:111

Fig. 2. a) Sample grammar G_{3323} automatically generated by the CFG generator. b) Sample individual that belongs to the language generated by G_{3323}.

Figure 2.a shows the sample grammar G_{3323} generated by the CFG generator for the values $n = 3$, $l = 3$, $p = 2$ and $c = 3$. Every individual generated by the grammar is composed of three parts: ORDER, CONNECTIONS and CLASS. The first part references to the order in which the layers of nodes are arranged. Insofar as the number of nodes and layers is the same in this example($n = l$), every layer contains only one node. The second part refers to the list of input connections of the n nodes in the individual, and the last part represents the input connections of the class node. The set of terminal symbols stores one symbol per node (except for the class node), a divider token ":" and numbers "1" and "0", which codify the presence or absence of a connection, respectively.

The first of the production rules displays the sequence of the three components of an individual. The second production rule determines an ordered list of nodes in the individual. The cardinality of the set of possible configurations of layers equals $l!$, which, in this example, is equal to $n!$ insofar as $n = l$. The third production rule codifies the input connections of every position in the ordered list separated by the divider token. Notice that the node occupying the first position in the list always receives zero input connections, as nodes can only be forward connected. For this reason, the codified connections start at the second position in the list, which can only receive one input connection from the first position. The second codified set of connections, after the divider token, refers to the second position in the list of nodes, which can receive input connections from positions one and two. The last production rule establishes that the class node may be connected with $c = 3$ nodes. Notice that connections refer to positions in the list of nodes, not actual nodes.

a) $G_{3212} = (\sum_N, \sum_T, S, P)$
$\sum_N$ = { S, ORDER, CONNECTIONS, CLASS }
$\sum_T$ = {0, 1, n_0, n_1, n_2, :}
P = {
S::= ORDER CONNECTIONS : CLASS
ORDER::= n_0 n_2 n_1 | n_1 n_0 n_2
CONNECTIONS::= 1:10 | 1:01 | 1:00 | 0:10 | 0:01 | 0:00
CLASS::= 110 | 101 | 100 | 011 | 010 | 001
}

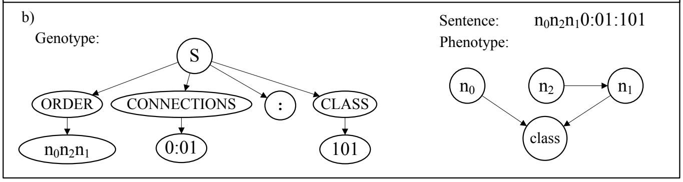


Fig. 3. a) Sample grammar G_{3212} automatically generated by the CFG generator. b) Sample individual that belongs to the language generated by G_{3212} .

Figure 2.b shows an individual that belongs to the language generated by G_{3323}. The derivation tree shown on the left is built by choosing the third consequent produced by "ORDER", that establishes the ordered list n_1, n_0, n_2. Then the third consequent produced by "CONNECTIONS" is chosen, codifying one input connection for the second node (n_0) coming from the first node (n_1), and another one for the third node (n_2) coming from the second node. Finally, the first consequent produced by "CLASS" fully connects the class node with n_0, n_1 and n_2. This derivation tree is the genotype that codifies the sentence shown in the upper-right of the diagram. The sentence codifies the phenotype of the individual, shown in the lower-right of the diagram.

Figure 3.a shows another sample grammar G_{3212} generated for the values $n = 3$, $l = 2$, $p = 1$ and $c = 2$. This grammar follows the same structure as presented in Figure 2.a. In this example, nodes ($n = 3$) are alternatively distributed across two layers ($l = 2$): $layer\,1 = \{n_0, n_2\}$ and $layer\,2 = \{n_1\}$. Insofar as only layers are reordered, the second rule produces only $\{n_0, n_2, n_1\}$ and $\{n_1, n_0, n_2\}$, a total of $l!$ orders. The third and fourth production rules limit the number of input connections to $p = 1$ for variable nodes, and to $c = 2$ for the class node, respectively. Like Figure 2.b, Figure 3.b shows the genotype, sentence and phenotype of an individual belonging to the language generated by G_{3212}.

The grammars in Figures 2.a and 3.a can be used to generate BN structures that solve classification problems with $n = 3$ feature variables. They differ in that the language generated by G_{3212} builds a smaller search space than the generated by G_{3323}, lowering the complexity of EvoBANE's evolutionary process.

2.2 The Fitness Calculator

Every individual is assigned a fitness score when it first enters the fitness calculator module. This score is a percentage measure that represents how accurately the BN codified by the individual (derivation tree) classifies a training set of instances. First, the fitness calculator decodifies the individual to output its BN structure (phenotype). Then, for each node in the network, a probabilistic estimator included in the Weka framework estimates its conditional probability table (CPT) [17]. Briefly, this estimator estimates the probability of every value in the node by counting the number of occurrences of that value within the training set. Once all the CPTs have been estimated, the fitness calculator uses the BN to classify the instances in the training set and calculates the fitness of the individual as the percentage of correctly classified instances.

3 Results

EvoBANE was used to classiffy two different datasets that belong to two different application domains extracted from the UCI repository [20]. The first one is called "Vote" and classifies voters as "Republicans" or "Democrats" considering sixteen feature variables. For this dataset, the CFG generator was set to generate the grammars $G_{16\,4\,3\,3}$, $G_{16\,3\,3\,3}$, and $G_{16\,1\,3\,3}$, to tackle the problem from three different angles. A genetic algorithm (GA) with a repair mechanism [17] was added as a fourth approach to compare its performance with EvoBANE. These four approaches were executed 20 times. Each execution is set up to evolve a population of 10 individuals for 100 generations.

Table 1 shows the descriptive statistics of the results of the four approaches in both training and test sets. The mean column shows that EvoBANE provides the best approaches in both the training and testing phases. The standard deviation column indicates that the GA always gets the same maximum fitness. This fitness is under the lower bound of the 95% confidence interval for the mean of the three EvoBANE approaches in both the training and testing phases. One possible explanation is that the genetic algorithm gets trapped in a local optimum with a fitness score equal to 92.7586, and is unable to explore the solution space as EvoBANE does.

An analysis of variance (ANOVA) test was performed in order to statistically compare the mean fitness of each of the three EvoBANE approaches. Table 2 details the results of the ANOVA test, where the null hypotheses (mean fitness

Table 1. Descriptive statistical results for 20 executions of EvoBANE with grammars $G_{16\,4\,3\,3}$, $G_{16\,3\,3\,3}$, $G_{16\,1\,3\,3}$ and a genetic algorithm for the "Vote" dataset

		N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
						Lower Bound	Upper Bound		
Fitness training	$G_{16\,4\,3\,3}$	20	95.8966	.2209	.0494	95.7931	96.0000	95.5172	96.2069
	$G_{16\,3\,3\,3}$	20	95.9828	.2313	.0517	95.8745	96.0910	95.5172	96.2069
	$G_{16\,1\,3\,3}$	20	96.1207	.6216	.1390	95.8298	96.4116	95.5172	98.6207
	GA	20	92.7586	.0000	.0000	92.7586	92.7586	92.7586	92.7586
	Total	80	95.1897	1.4557	.1627	94.8657	95.5136	92.7586	98.6207
Fitness testing	$G_{16\,4\,3\,3}$	20	96.7586	.5526	.1236	96.5000	97.0173	95.8621	97.2414
	$G_{16\,3\,3\,3}$	20	96.5517	.5003	.1119	96.3176	96.7859	95.8621	97.2414
	$G_{16\,1\,3\,3}$	20	96.3448	.9253	.2069	95.9118	96.7779	93.1034	97.2414
	GA	20	95.8621	.0000	.0000	95.8621	95.8621	95.8621	95.8621
	Total	80	96.3793	.6720	.0751	96.2298	96.5289	93.1034	97.2414

Table 2. Results of the ANOVA test for the "Vote" dataset

		Sum of Squares	df	Mean Square	F	Sig.
Fitness training	Between Groups	.511	2	.256	1.569	.217
	Within Groups	9.287	57	.163		
	Total	9.798	59			
Fitness testing	Between Groups	1.712	2	.856	1.819	.171
	Within Groups	26.825	57	.471		
	Total	28.537	59			

values are equal in the training and test sets) cannot be rejected with a significance of the value $F(df = 2/57)$ equal to 0.217 in training and 0.171 in the testing. Even though the null hypotheses cannot be rejected, the significance values indicate that there is a high probability of finding differences between the means. Tukey's HSD test has been used to show where the differences between the three approaches lie.

Table 3 contains the results of Tukey's HSD test. It shows that the means of the first two approaches are closer to each other than to the third approach in both the training and testing phases. Insofar as the lowest fitness during the testing phase belongs to the third approach, we infer that the simplification of the search space by the grammar $G_{16\,1\,3\,3}$ leads to slight drop in the accuracy of the BNs generated by EvoBANE.

The second dataset is called "Postoperative", and its classification task is to determine where patients in a postoperative recovery area should be sent to next: intensive care unit, hospital floor or home. Eight feature variables should be used for classification purposes. The CFG generator was set to generate the grammars G_{8444}, G_{8344} and G_{8144}. Table 4 shows the descriptive statistics of these three EvoBANE approaches and the GA. All the EvoBANE approaches again achieve better results than the GA in both the training and the testing phases.

Table 5 details the results of the ANOVA test performed on the three EvoBANE approaches. In this dataset, the equality of the means (null hypothesis) cannot be rejected in the testing phase with a significance of the value $F(df = 2/57)$ equal to 0.886. This suggests that all three EvoBANE approaches achieve the same results.

Table 3. Results of Tukey's HSD post-hoc test for the "Vote" dataset

Dependent Variable	(I) Approach	(J) Approach	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
						Lower Bound	Upper Bound
Fitness training	G_{16433}	G_{16333}	-.08621	.1105	.863	-.3766	.2042
		G_{16133}	-.22414	.1105	.187	-.5145	.0662
	G_{16333}	G_{16433}	.08621	.1105	.863	-.2042	.3766
		G_{16133}	-.13793	.1105	.599	-.4283	.1524
	G_{16133}	G_{16433}	.22414	.1105	.187	-.0662	.5145
		G_{16333}	.13793	.1105	.599	-.1524	.4283
Fitness testing	G_{16433}	G_{16333}	.20690	.1879	.690	-.2866	.7004
		G_{16133}	.41379	.1879	.132	-.0797	.9073
	G_{16333}	G_{16433}	-.20690	.1879	.690	-.7004	.2866
		G_{16133}	.20690	.1879	.690	-.2866	.7004
	G_{16133}	G_{16433}	-.41379	.1879	.132	-.9073	.0797
		G_{16333}	-.20690	.1879	.690	-.7004	.2866

Table 4. Descriptive statistical results for 20 executions of EvoBANE with grammars $G_{8\,4\,4\,4}$, $G_{8\,3\,4\,4}$, $G_{8\,1\,4\,4}$ and a genetic algorithm for the "Postoperative" dataset

		N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
						Lower Bound	Upper Bound		
Fitness training	G_{8444}	20	80.00	.000	.000	80.00	80.00	80	80
	G_{8344}	20	81.50	4.617	1.032	79.34	83.66	80	95
	G_{8144}	20	80.00	.000	.000	80.00	80.00	80	80
	GA	20	75.00	.000	.000	75.00	75.00	75	75
	Total	80	79.13	3.354	.375	78.38	79.87	75	95
Fitness testing	G_{8444}	20	72.6667	6.3614	1.4225	69.6894	75.6439	63.3333	80
	G_{8344}	20	73.6667	7.0835	1.5839	70.3515	76.9818	63.3333	80
	G_{8144}	20	73.5000	7.1308	1.5945	70.1627	76.8373	63.3333	80
	GA	20	60	.0000	.0000	60	60	60	60
	Total	80	69.9583	8.2249	.9196	68.1280	71.7887	60	80

Table 5. Results of the ANOVA test for the "Postoperative" dataset

		Sum of Squares	df	Mean Square	F	Sig.
Fitness training	Between	30.000	2	15.000	2.111	.130
	Within Groups	405.000	57	7.105		
	Total	435.000	59			
Fitness testing	Between	11.481	2	5.741	.122	.886
	Within Groups	2688.333	57	47.164		
	Total	2699.815	59			

Therefore the reduction of the search space by the grammars G_{8344} and G_{8144} does not affect the performance of the BNs generated by EvoBANE.

4 Conclusions

The EvoBANE system has been presented as an evolutionary approach that automatically generates Bayesian networks for solving classification problems.

EvoBANE uses a general-purpose grammar-guided programming core that implements a complete evolutionary mechanism (initialization, selection, crossover and replacement) whose grammatical crossover operator avoids the closure problem.

A CFG generator has been implemented in EvoBANE in order to automate the creation of context-free grammars that generate languages whose sentences codify valid Bayesian network structures that preserve DAG constraints. The CFG generator inputs the specifications of the classification problem to solve and the features of the solutions to be generated, and outputs the CFG that EvoBANE uses to generate and evolve a population of Bayesian networks.

The results show that the Bayesian networks automatically generated by EvoBANE accurately solve classification problems from two different application domains. In these two cases EvoBANE has been able to explore the search space, avoiding local optima and reaching good Bayesian networks without a repair mechanism. The flexibility of the CFG generator vests EvoBANE with the capability to modify the search space in order to find the simplest Bayesian network that solves the problem.

Acknowledgements

This work has been supported by the LIA-Group (http://www.lia.upm.es/) of the Universidad Politécnica de Madrid.

References

1. Wong, M.L., Guo, Y.Y.: Learning bayesian networks from incomplete databases using a novel evolutionary algorithm. Decision Support Systems 45, 368–383 (2008)
2. Darwiche, A.: Bayesian networks. Communications of the ACM 53(12), 80–90 (2010)
3. Niedermayer, D.: An Introduction to Bayesian Networks and Their Contemporary Applications. In: Innovations in Bayesian Networks. SCI, pp. 117–130. Springer, Heidelberg (2008)
4. Chen, X., Anantha, G., Lin, X.: Improving bayesian network structure learning with mutual information-based node ordering in the k2 algorithm. IEEE Transactions on Knowledge and Data Engineering 20(5), 1–13 (2008)
5. Larrañaga, P., Poza, M., Yurramendi, Y., Murga, R.H., Kuijpers, C.M.H.: Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(9), 912–926 (1996)
6. Barriére, O., Lutton, E., Wuillemin, P.: Bayesian network structure learning using cooperative coevolution. In: GECCO 2009, Montréal, Canada, July 2009, pp. 755–762 (2009)
7. Alonso-Barba, J.I., delaOssa, L., Puerta, J.M.: Structural learning of bayesian networks by using variable neighbourhood search based on the space of orderings. In: Ninth Conference on Intelligent Systems Design and Applications, pp. 1435–1440 (2009)

8. Font, J.M., Manrique, D., Ríos, J.: Evolutionary construction and adaptation of intelligent systems. Expert Systems with Applications 37, 7711–7720 (2010)
9. Koza, J.: Genetic programming: On the programming of computers by means of natural selection. The MIT Press, Cambridge (1992)
10. Dong, L., Liu, G., Yuan, S., Li, Y., Li, Z.: Classifier learning algorithm based on genetic algorithms. In: ICICIC 2007 Second International Conference on Innovative Computing, Information and Control, September 2007, pp. 126–129 (2007)
11. Larrañaga, P., Kuijpers, C.M.H., Murga, R.H., Yurramendi, Y.: Learning bayesian network structures by searching for the best ordering with genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans 26(4), 487–493 (1996)
12. De Stefano, C., Fontanella, F., Scotto di Freca, A., Marcelli, A.: Learning bayesian networks by evolution for classifier combination. In: 10th International Conference on Document Analysis and Recognition, pp. 966–970 (2009)
13. Davidson, C.: Identifying gene regulatory networks using evolutionary algorithms. Journal of Computing Sciences in Colleges 25(5), 231–237 (2010)
14. Wong, M.L., Lee, Y.L., Leung, K.S.: A hybrid approach to learn bayesian networks using evolutionary programming. In: WCCI 2002 Proceedings of the 2002 World Congress on Computational Intelligence, vol. 2, pp. 1314–1319 (2002)
15. Lee, J., Chung, W., Kim, E.: Structure learning of bayesian networks using dual genetic algorithm. IEICE Transactions on Informations and Systems E91-D(1), 32–43 (2008)
16. Palacios-Alonso, M.A., Brizuela, C.A., Sucar, L.E.: Evolutionary learning of dynamic naive bayesian classifiers. Journal on Automated Reasononing 45, 21–37 (2010)
17. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explorations 11(1), 10–18 (2009)
18. Font, J.M., Manrique, D.: Grammar-guided evolutionary automatic system for autonomously building biological oscillators. IEEE Congress on Evolutionary Computation, 1–7 (July 2010)
19. Couchet, J., Manrique, D., Ríos, J., Rodríguez-Patón, A.: Crossover and mutation operators for grammar-guided genetic programming. Soft Computing: A Fusion of Foundations, Methodologies and Applications 11(10), 943–955 (2007)
20. Frank, A., Asuncion, A.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2010), http://archive.ics.uci.edu/ml

A Novel Linear Cellular Automata-Based Data Clustering Algorithm

Javier de Lope[1,2] and Darío Maravall[1]

[1] Cognitive Robotics Group
Dept. of Artificial Intelligence
Universidad Politécnica de Madrid
[2] Dept. Applied Intelligent Systems
Universidad Politécnica de Madrid
javier.delope@upm.es, dmaravall@fi.upm.es

Abstract. In this paper we propose a novel data clustering algorithm based on the idea of considering the individual data items as cells belonging to an uni-dimensional cellular automaton. Our proposed algorithm combines insights from both social segregation models based on Cellular Automata Theory, where the data items themselves are able to move autonomously in lattices, and also from Ants Clustering algorithms, particularly in the idea of distributing at random the data items to be clustered in lattices. We present a series of experiments with both synthetic and real datasets in order to study empirically the convergence and performance results. These experimental results are compared to the obtained by conventional clustering algorithms.

Keywords: Cellular Automata, Machine Learning, Pattern Recognition, Data Mining, Data Clustering, Social Segregation Models, Ants Clustering.

1 Introduction

Clustering is a method of unsupervised learning used in different fields such as Machine Learning, Data Mining, Pattern Recognition, and so on. It deals with the assignment of a set of data items into subsets or *clusters* in such a way that the data items in the same cluster are *similar*.

Our proposed algorithm is inspired on social segregation models based on Cellular Automata Theory. In this line of thought Schelling [12] originally proposed models of segregation in which the population is composed of two well differentiated types of individuals. Each individual cares about the neighbors. If the neighbors *dissatisfied* him, then the individual moves to the nearest point that meets his minimum demand, i.e. the nearest point at which a significant proportion of his neighbors satisfy his demands.

In [7] Hegselmann *et al.* also present similar ideas based on the use of Cellular Automata Theory [10] by identifying the individuals with cells in a toroidal rectangular lattice and the individuals' choices with the states that the cells

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 70–79, 2011.

can evolve. Under this approach, social phenomena such as the emergence of order, the micro-macro relations and the social dynamics can be analytically understood with the help of Cellular Automata Theory.

Several data clustering algorithms based on the social collective behavior have been proposed [11,9] and also in other different tasks [8,14,6,5].

Beckers *et al.* [1] run a classic experiment in which a group of mobile robots gather several randomly distributed objects and cluster them into separated piles. The agents' coordination in those experiments was achieved through *stigmergy*, principle originally developed for the description of termite building behavior. The agents' behavior is determined by an indirect communication between them, which is carried out sensing and modification of the agents' local environment. This situated experiment inspired several successful data clustering algorithms [2,15,13].

In ants clustering algorithms, a group or colony of ant-like agents perform a clustering task in a toroidal 2D rectangular lattice or grid in which the N-dimensional data items to be clustered have been previously scattered at random with a single data item at each site. The individual data items randomly scattered in the rectangular lattice can be picked-up, transported and dropped by the ants-like agents. The two basic actions of picking-up and dropping a data item are performed in a probabilistic way by the agents. Generally speaking, the probability for an agent of picking-up a particular data item depends proportionally on the number of similar data items deposited in its neighborhood, so that the lower the number of similar neighbors, the higher the probability of picking-up a particular data item. Conversely, the higher the number of similar neighbors, the higher the probability for a loaded agent to drop a data item at an empty site in the toroidal rectangular grid.

In this paper we propose a novel data clustering algorithm based on the idea of considering the individual data items as mobile agents. Our proposed algorithm combines insights from both ants clustering algorithms —particularly in the idea of distributing at random the data items to be clustered toroidal discrete lattices— and also from social segregation models based on Cellular Automata Theory, in which the data items themselves, like individuals in a social neighborhood, are able to move autonomously in the toroidal linear lattice, following a basic rule for staying at a particular site if a similarity with the neighbors' condition holds or for migrating from a particular location in which the similarity with the neighbors' condition does not hold.

2 One-Dimensional Cellular Automata-Based Clustering

In this paper we propose to employ one-dimensional discrete lattices of cells upon which cellular automata operate performing an unsupervised data clustering process. Each cell $\sigma_{i \in \mathcal{L}}(t)$ in the discrete lattice $\mathcal{L}$ is the i^{th} cell at time t and is related to a particular data item, x_i, in the dataset. The number of cells in the lattice, N, is also the number of data items in the dataset. As usual for one-dimensional lattices we are assuming periodic boundary conditions, in which σ_{i+1} is identified with σ_i.

At initialization ($t = 0$), the data items x_i are randomly associated to the cells $\sigma_{i \in \mathcal{L}}(t = 0)$. Then, a set of transition rules ϕ_r is iteratively applied on a range of r cells (r specifies the size of the neighborhood for the rule ϕ_r). The transitions are local in both space and time: a cell evolves according a function of the current state of that cell and its neighboring cells. One iteration step is achieved by the application of the rules ϕ_r to each cell in the lattice $\mathcal{L}$. The number of rules in the set, R, is computed in terms of the number of cells in the lattice, N, and the size of the greater cluster: the greater R, the greater the clusters sizes can be.

The process finishes when the the lattice $\mathcal{L}$ converges to a final state, in which the states of each cell $\sigma_i(t)$ are *stationary* and *equivalent* to the previous states $\sigma_i(t-1)$, for each $i = 1, \ldots, N$. Two states σ_i and σ_j are equivalent when the associated data items x_i and x_j belong to the same cluster.

A generic rule ϕ_r for a neighborhood of r cells can be written as follows:

$$[\sigma_i(t+1), \sigma_{i+1}(t+1), \ldots, \sigma_{i+r-1}(t+1)] = \phi_r\left(\sigma_i(t), \sigma_{i+1}(t), \ldots, \sigma_{i+r-1}(t)\right) \tag{1}$$

where i is the initial or *reference cell.* Notice that r must be greater or equal to 2, i.e. the minimum neighborhood considered has 3 cells. Fig. 1 shows one iteration of the algorithm.

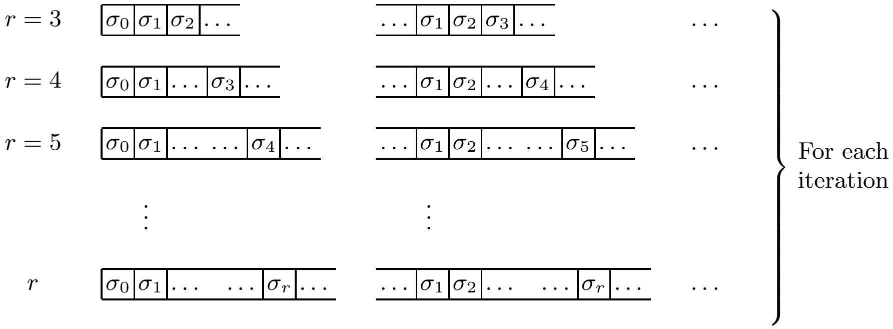

$r = 3$
σ_0 σ_1 σ_2 ...
... σ_1 σ_2 σ_3 ...
...
$r = 4$
σ_0 σ_1 ... σ_3 ...
... σ_1 σ_2 ... σ_4 ...
...
$r = 5$
σ_0 σ_1 ... ... σ_4 ...
... σ_1 σ_2 ... ... σ_5 ...
...
r
σ_0 σ_1 ... ... σ_r ...
... σ_1 σ_2 ... ... σ_r ...
...
For each iteration

Fig. 1. For each iteration the rules ϕ_r are sequentially applied to each cell in the lattice $\mathcal{L}$ and r is also increased for checking if the data items are similar to the current reference cell

The rule ϕ_r computes the distance between the feature vectors of the data items associated to the reference cell $\sigma_i(t)$ and its contiguous cell $\sigma_{i+1}(t)$ and compares it to the distance between the feature vectors of the data items associated to the reference cell $\sigma_i(t)$ and the last cell in that neighborhood $\sigma_{i+r-1}(t)$. The data items associated to the cell $\sigma_{i+1}(t+1)$ will be the data items with the lesser distance. Formally:

$$\sigma_{i+1}(t+1) = \begin{cases} \sigma_{i+1}(t) & \text{if } d(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}) < d(\boldsymbol{x}_i, \boldsymbol{x}_{i+r-1}) \\ \sigma_{i+r-1}(t) & \text{otherwise} \end{cases} \tag{2}$$

Equally, the data items associated to the cell $\sigma_{i+r-1}(t+1)$ will be the data item with the greater distance:

$$\sigma_{i+r-1}(t+1) = \begin{cases} \sigma_{i+r-1}(t) & \text{if } d(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}) < d(\boldsymbol{x}_i, \boldsymbol{x}_{i+r-1}) \\ \sigma_{i+1}(t) & \text{otherwise} \end{cases} \tag{3}$$

These rules keep together data with similar values in the feature vectors. Fig. 2 depicts the states which the tape can evolve depending on the data items considered.

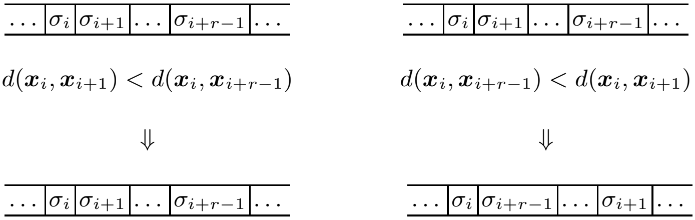

$\ldots \mid \sigma_i \mid \sigma_{i+1} \mid \ldots \mid \sigma_{i+r-1} \mid \ldots$
$d(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}) < d(\boldsymbol{x}_i, \boldsymbol{x}_{i+r-1})$
$\Downarrow$
$\ldots \mid \sigma_i \mid \sigma_{i+1} \mid \ldots \mid \sigma_{i+r-1} \mid \ldots$
$\ldots \mid \sigma_i \mid \sigma_{i+1} \mid \ldots \mid \sigma_{i+r-1} \mid \ldots$
$d(\boldsymbol{x}_i, \boldsymbol{x}_{i+r-1}) < d(\boldsymbol{x}_i, \boldsymbol{x}_{i+1})$
$\Downarrow$
$\ldots \mid \sigma_i \mid \sigma_{i+r-1} \mid \ldots \mid \sigma_{i+1} \mid \ldots$

Fig. 2. A tape state can evolve to two different states depending on the distances between data item associated to the reference cell σ_i and the cells σ_{i+1} and σ_{i+r-1}

By increasing systematically the range r of the rules, the number of contiguous cells belonged to data items on the same cluster also increases. After a few iterations, the system converges to a final state where all the cells associated to data items of the same cluster are contiguous. The items in each cluster can be easily detected by comparing the distance between each cell and the next one (i.e. the chainmap diagram). When that distance is much greater than the previous ones, the new cell corresponds to a new cluster.

3 Experimental Results

We have tested the proposed algorithm with both synthetic and real data like the well-known Iris dataset. In the sequel we describe and comment the results.

3.1 Clustering Results with Synthetic Data

The synthetic dataset used for experimental purposes is shown in Fig. 3. It is composed of 4 two-dimensional classes. Each class contains 20 data items.

After initializing the cellular automata linear lattice by depositing at random the individual data items, one at each cell or site, the cellular automata rules explained in section 2 start to act.

Fig. 4 displays the successive states of the automata lattice. Each line shows the tape's state in each iteration. We have assigned a different grey level to each cluster shown in the Fig. 3: black corresponds to the cluster represented with circles, dark grey to the triangles, light grey to the stars and white to the squares.

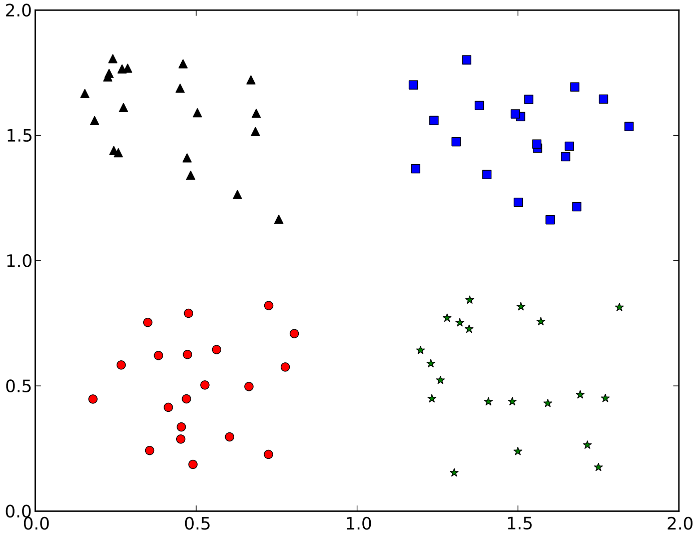


Fig. 3. Synthetic dataset used for experimental purposes

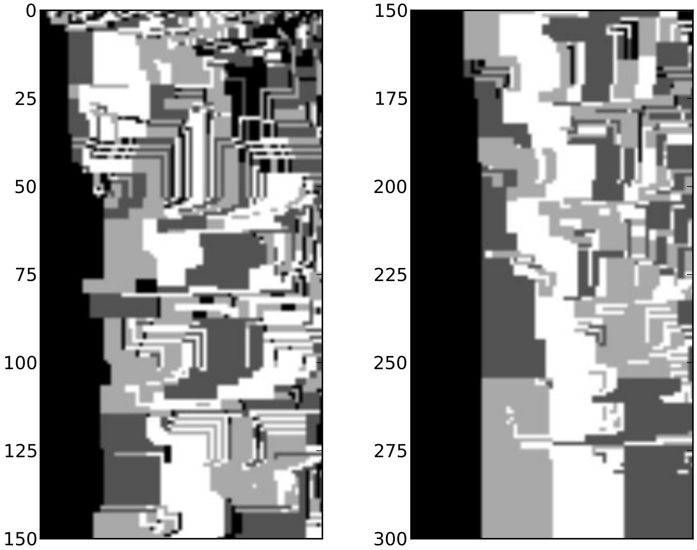


Fig. 4. Successive states of the automata lattice for the synthetic dataset. Each grey level represents a different cluster.

In the initial iterations, which are shown in the left side, the data items keep the initial random order. According the algorithm operates, the data items begin to keep together and the clusters start to appear. Around the 75 iteration — approximately in the middle of the left diagram— the black and greys clusters are almost clustered, just remain some data items belonged to these clusters in the right side. The white cluster is divided into two noncontiguous clusters. Around the 200 iteration the black cluster is completely formed and keeps in such state until the end. The final tape's state, in which the data items are grouped into the 4 existing clusters, is achieved in the iteration 283. Once the convergence is reached, this state is maintained.

Afterwards an important post-processing step oriented to the automatic grouping of the data items scattered on the tape has to be applied.

As the data items are linearly grouped in the cellular automaton tape a straightforward way of finding the natural clusters within the data is by analyzing the chainmap diagram formed by the distances of the successive data items. The main problem is to detect automatically the optimum threshold that gives the correct number of clusters for each dataset.

Fig. 5 displays the chainmap of the synthetic dataset in which it can be notice the existence of 4 local maxima, each of one corresponding to an individual cluster. These local maxima are located in the 20, 40, 60 and 80 bins. Notice that the last one is needed for defining the the frontier between the last cluster and the first one due to we are considering a toroidal lattice.

We have also employed the standard k-means clustering with this synthetic dataset and we have achieved similar successful results for $k = 4$. As in our proposed algorithm we have used the Euclidean distance as metric.

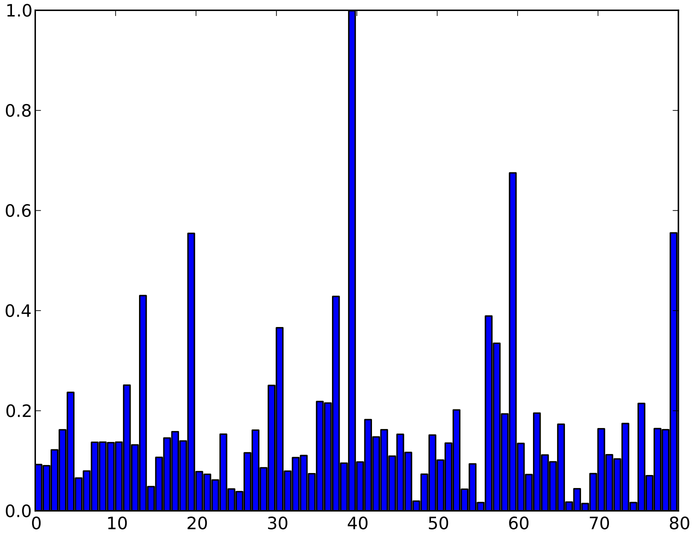


Fig. 5. Chainmap of the synthetic dataset

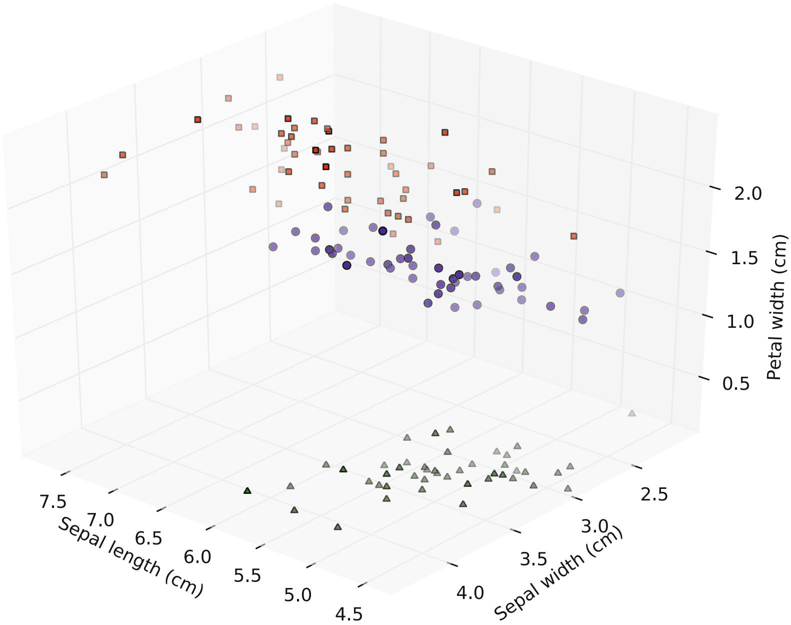


Fig. 6. The Iris dataset. The triangle marks represent the Iris setosa, the circles correspond to the Iris versicolor and the squares are used for the Iris virginica.

3.2 Clustering Results with the Iris Dataset

We have also tested the one-dimensional cellular automata-based clustering algorithm with a real dataset as the Iris dataset. The Iris dataset [4] was first used and even created by Fisher [3] in his pioneering research work on linear discriminant analysis, and today it is still an up-to-date, standard pattern recognition problem for testing discriminant techniques and algorithms.

In this well-known and classical multiclass pattern recognition problem, three classes of Iris flowers (*setosa*, *versicolor* and *virginica*) have to be classified according to four continuous discriminant variables measured in centimeters: sepal length, sepal width, petal length and petal width.

Fig. 6 shows the three classes. We have only represented three variables of this four-dimensional dataset: the sepal length, the sepal width and the petal width. The triangle marks represent the Iris setosa, the circles are the Iris versicolor items and the squares correspond to the Iris virginica.

It is well-known that this dataset only contains two clusters with an obvious separation. The Iris setosa is in one of those clusters while the other two species, Iris versicolor and Iris virginica, are in the other cluster.

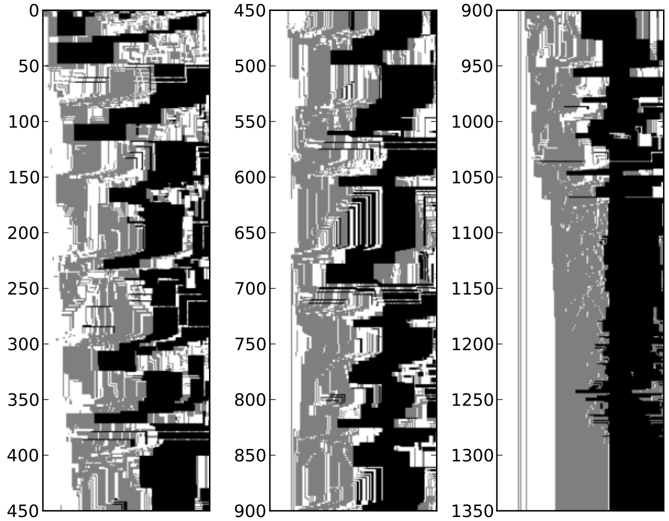


Fig. 7. Successive states of the automata lattice for the Iris dataset

Fig. 7 shows the successive states of the automata lattice. As in the synthetic case each line represents the tape's line in each iteration. As the Iris dataset contains three classes we have only employed the black, grey and white colors to represent the Iris setosa, Iris versicolor and Iris virginica data items, respectively.

The final tape's state is achieved in the iteration 1291 although it is not obtained a perfect clustering in this case. Only the black zone which represents the Iris setosa shows a compact state. Some individuals corresponding to the Iris versicolor (grey cluster) are included in the middle of the Iris virginica zone (white cluster) and vice versa.

This fact can be clearly observed in the matching matrix shown in Table 1. All the 50 Iris setosa items are classified in the correct cluster, 3 Iris versicolor items are erroneously classified as Iris virginica, and 1 Iris virginica item is erroneously classified as Iris versicolor.

Table 1. Iris dataset matching matrix obtained with the proposed algorithm

	Iris setosa	Iris versicolor	Iris virginica
Iris setosa	50	0	0
Iris versicolor	0	47	3
Iris virginica	0	1	49

The obtained results are quite comparable to the ones obtained with a conventional k-means algorithm $k = 3$. The resulting matching matrix is shown in Table 2.

Table 2. Iris dataset matching matrix obtained with k-means using the same metric than in our proposed algorithm, i.e. Euclidean distance

	Iris setosa	Iris versicolor	Iris virginica
Iris setosa	50	0	0
Iris versicolor	0	48	2
Iris virginica	0	14	36

Fig. 8 shows the chainmap of the Iris dataset. In this case it can be clearly distinguished 3 local maxima, each of one corresponding to an individual cluster.

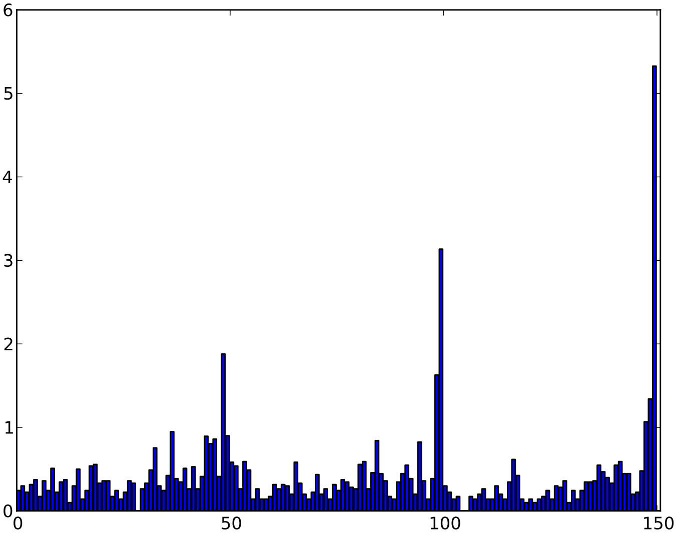


Fig. 8. Chainmap of the Iris dataset

4 Conclusions and Further Work

A novel algorithm for data clustering based on linear cellular automata has been proposed. The method identifies the individual data items as cells belonging to an uni-dimensional cellular automaton and it is inspired in both social segregation models and also Ant Clustering algorithms.

The results obtained as synthetic as real datasets improve significantly the ones obtained with conventional unsupervised methods such as the k-means algorithm.

Although the data items are correctly ordered in the tape, still remains the post-processing task of finding the real clusters of the dataset. As we have used chainmaps diagrams formed by the distances of the successive data items, that issue corresponds to analyze the diagram and to select the optimum threshold that gives the correct clustering solution for each particular dataset.

References

1. Beckers, R., Holland, O.E., Deneubourg, J.L.: From local actions to global tasks: stigmergy and collective robotics. In: Brooks, R., Maes, P. (eds.) Artificial Life IV, pp. 181–189. The MIT Press, Cambridge, MA (1994)
2. Chen, L., Xu, X.H., Chen, Y.X.: An adaptive ant colony clustering algorithm. In: Proc. 3rd Int. Conf. on Machine Learning and Cybernetics, pp. 1387–1392 (2004)
3. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annual Eugenics, 7, Part II, 179–188 (1936)
4. Frank, A., Asuncion, A.: UCI Machine Learning Repository (2010), http://archive.ics.uci.edu/ml
5. Gallego, J., Hernández, C., Graña, M.: A morphological cellular automata based on morphological independence. Logic Journal of the IGPL
6. Ganguly, N., Sikdar, B., Deutsch, A., Canright, G., Chaudhuri, P.: A survey on cellular automata. Tech. rep (2003)
7. Hegselmann, R.: Modeling social dynamics by cellular automata. In: Liebrand, W.B.G., Nowak, A., Hegselmann, R. (eds.) Computer Modeling of Social Processes, pp. 37–64. SAGE Publications, London (1998)
8. Ilachinski, A.: Cellular Automata. A discrete universe. World Scientific, Singapore (2001)
9. Sree, P.K., Raju, G.V.S., Babu, I.R., Raju, S.V.: Improving quality of clustering using cellular automata for information retrieval. Journal of Computer Science 4(2), 167–171 (2008)
10. Neumann, J.V.: Theory of Self-reproducing Autamata (edited and completed by Arthur W. Burks). University of Illinois Press, Urbana (1966)
11. Saha, S., Maji, P., Ganguly, N., Roy, S., Chaudhuri, P.P.: Cellular automata based model for pattern clustering. In: Proc. 5th Int. Conf. on Advances in Pattern Recognition, pp. 122–126 (2003)
12. Schelling, T.C.: Dynamic models of segregation. Journal of Mathematical Sociology 1(2), 143–186 (1971)
13. Moere, A.V., Clayden, J.J., Dong, A.: Data clustering and visualization using cellular automata ants. In: Sattar, A., Kang, B.-h. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 826–836. Springer, Heidelberg (2006)
14. Wolfram, S.: A new kind of science. Wolfram Media, Inc., Champaign (2002)
15. Xu, X., Chen, L., He, P.: Ant clustering embeded in cellular automata. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) ECAL 2005. LNCS (LNAI), vol. 3630, pp. 562–571. Springer, Heidelberg (2005)

Probabilistic versus Incremental Presynaptic Learning in Biologically Plausible Synapses

Francisco Javier Ropero Peláez[1] and Diego Andina[2]

[1] Center for Mathematics, Computation and Cognition, Federal University of ABC
francisco.pelaez@ufabc.edu.br
[2] Group for automation in signals and communication. UPM, Spain
d.andina@upm.es

Abstract. In this paper, the presynaptic rule, a classical rule for hebbian learning, is revisited. It is shown that the presynaptic rule exhibits relevant synaptic properties like synaptic directionality, and LTP metaplasticity (long-term potentiation threshold metaplasticity). With slight modifications, the presynaptic model also exhibits metaplasticity of the long-term depression threshold, being also consistent with Artola, Brocher and Singer's (ABS) influential model. Two asymptotically equivalent versions of the presynaptic rule were adopted for this analysis: the first one uses an incremental equation while the second, conditional probabilities. Despite their simplicity, both types of presynaptic rules exhibit sophisticated biological properties, specially the probabilistic version.

Keywords: Metaplasticity, ABS rule, NMDA channel, BCM rule.

1 Introduction

Synapses are neural structures that modulate presynaptic activity, converting this activity into a higher or lower postsynaptic activation (voltage). The magnitude that relates postsynaptic voltage with presynaptic activity is called synaptic efficiency or synaptic weight.

To model brain learning, it is necessary to start with a plausible hypothesis of synaptic weight modification. According to the plasticity curve in Figure 1.a, synaptic modification depends on the value "a" of postsynaptic activation [1]. If the value (voltage) of this postsynaptic depolarization is above the Long-Term Potentiation (LTP) threshold (white dot), a positive variation of synaptic weight, that is, synaptic potentiation, occurs. If the value of postsynaptic activation is between the LTP threshold and the Long-Term Depression (LTD) threshold (black dot), a negative variation of synaptic weight, that is, synaptic depression, takes place. Below the LTD threshold, neither potentiation nor depression occurs.

Another important property related to synaptic plasticity is metaplasticity [2] which is a homosynaptic property. Metaplasticity is related to the change in the shape of the plasticity curve by either a change of the regime (average over

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 80–89, 2011.

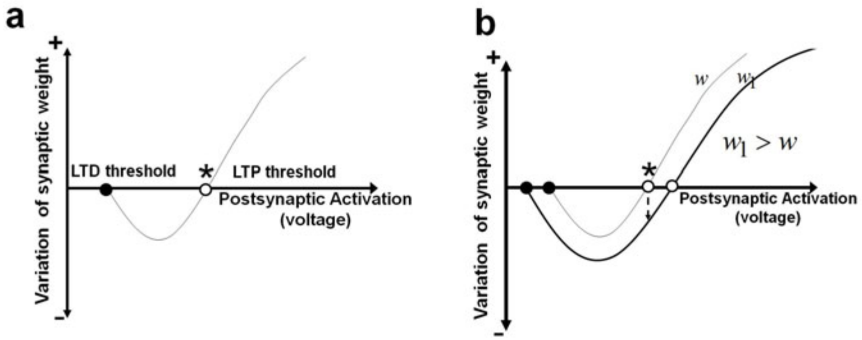

a
Variation of synaptic weight
+
-
LTD threshold
*
LTP threshold
Postsynaptic Activation (voltage)
b
Variation of synaptic weight
+
-
w
w1
$w_1 > w$
*
Postsynaptic Activation (voltage)

Fig. 1. Synaptic weight modification of a biological synapse according to its postsynaptic activation. (a) For values lower than the LTD threshold (black dot) no variation of synaptic weight takes place. For values between the LTP threshold (white dot) and the LTD threshold, the variation of the synaptic weights is negative, i.e. the synaptic weight decreases. For values higher than the LTP threshold, the synaptic weight increases. (b) Synaptic metaplasticity: Curves corresponding to higher values of ω have a higher LTP threshold and a lower LTD threshold.

time) of postsynaptic activations $\hat{a}$, or a concomitant change in synaptic weight ω (Fig. 1.b).

Let us consider the curve of figure 1.a in which the independent variable is the postsynaptic activation and where other parameters such as $\hat{a}$ or ω are kept constant. For example, let us evaluate the variation of weight for a specific value of postsynaptic activation by setting the postsynaptic activation (see asterisk in the curve of Fig. 1.a) to the value of the LTP threshold. In this case, no variation of weight is observed. However, if the regime of synaptic activations $\hat{a}$ is changed, so that ω reaches a higher value ω_1, metaplasticity is manifested by the modification of the shape of the curve as shown in figure 1.b. In this new situation, if the same postsynaptic voltage is applied as before (see asterisk in the curve), instead of obtaining a null variation of weight, the new curve yields a negative variation (a decrement) of weight (see arrow).

In the first articles that study metaplasticity [2], metaplasticity was related to the rightward shift of LTP-thresholds for higher $\hat{a}$'s or ω's. More recent studies [3] showed that the LTD thresholds diminish in the same circumstances (Figure 1.b). In summary, once synapses are positively primed (i.e. there is an increment in weight), the interval between thresholds broadens, thereby favouring subsequent synaptic depression.

Along the years, different mathematical models of synaptic computation have been proposed (see Figure 2.a). In the classical Hebb model [4], the curve relating the increment of synaptic weight to postsynaptic activation is a straight line without synaptic depression. In Sejnowski's covariance model [5][6], regions of potentiation and depression are separated by a LTP threshold (white dot).

Artola, Brocher and Singer's [8] extended model (ABS model, Figure 2.b) is not analytical, as those just discussed, but is based on empirical experimental

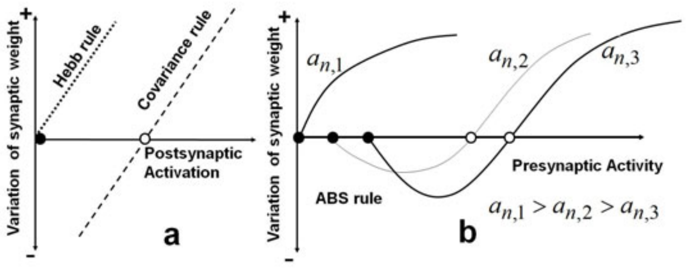


Fig. 2. Different models of synaptic plasticity and metaplasticity. (a) Hebb rule and Sejnowski's covariance rule. (b) The ABS qualitative model by Artola, Brocher and Singer. a_n, represents the postsynaptic activation due to neighbouring synapses to the one under consideration. For higher values of a_n, the curve is shifted to the left.

data. In the ABS model, LTP and LTD thresholds shift to lower values for higher levels of the activation of neighbouring synapses a_n. For example, if the presynaptic activity (action potential frequency) of a specific synapse is negligible, and activations from neighbouring synapses are high, as in curve $a_{n,1}$, synaptic potentiation occurs in that synapse.

In the following sections, the incremental and probabilistic versions of the presynaptic rule [9] will be analyzed in order to assess their biological plausibility in the light of the mentioned synaptic properties (i.e. metaplasticity of both LTP and LTD thresholds, and the ABS rule).

In the incremental version of the presynaptic rule, the variation of the synaptic weight ω is calculated as:

$$\triangle \omega = \xi I(O - \omega) \tag{1}$$

where I and O (in uppercase) are the presynaptic and postsynaptic action potential frequencies or, if normalized, the presynaptic and postsynaptic action potential probabilities respectively. ξ is a small positive constant. According to Grossberg[9] and Minai [10] this incremental version of the presynaptic rule is asymptotically equivalent to the following probabilistic expression:

$$\omega = P(\frac{o}{i}) \tag{2}$$

which is regarded to be "the probabilistic version of the presynaptic rule". $P(\frac{o}{i})$ is the conditional probability of a postsynaptic action potential, o, given a presynaptic action potential i (notice that binary action potentials are represented in lowercase while action potential frequencies or probabilities are in uppercase).

2 Methods

The two versions of the presynaptic rule correspond respectively to the following mathematical models:

In the first model a rate-code neuron is used, in which the output of the neuron is a number expressing its firing frequency or probability. In this model the incremental rule of synaptic plasticity is applied.

In the latter, a spiking neuron model is used, in which the output is binary, and triggers according to the calculated firing probability. In this case, the probabilistic version of the presynaptic rule is applied.

In the first rate-code model, the action potential probabilities of presynaptic and postsynaptic neurons are respectively written I and O (in upper case, as explained before). The synaptic weight ω relates I and the excitatory postsynaptic potential, e, at a specific synapse j.

$$e_j = \omega_j I_j \tag{3}$$

The neuron's activation, a, is obtained after summing the postsynaptic potentials of all synapses:

$$a = \sum_{j=1}^{n} e_j = \sum_{j=1}^{n} \omega_j I_j \tag{4}$$

Afterwards, the action potential probability is calculated as a non-linear (sigmoidal) function of a: $O = f(a)$.

The rate code model is simplified by working in the linear part of the sigmoidal curve.

In the second mathematical model, the so called spiking output model, synaptic weights are calculated taking into account the degree of correlation between presynaptic and postsynaptic action potentials considered as binary outputs. When an action potential is triggered it is represented by a binary one, whereas the absence of an action potential is represented by a binary zero. The output of the presynaptic neuron ($bit = 1$ or $bit = 0$) is denoted by i (lowercase) while the postsynaptic output ($bit = 1$ or $bit = 0$) is denoted by o. The probabilistic version of the presynaptic rule for obtaining synaptic weights is:

$$\omega = P(\frac{o}{i}) = \frac{n(o \cap i)}{n(i)} \tag{5}$$

where *n( )* in the numerator counts the number of coincidences between presynaptic and postsynaptic binary outputs.

The calculus of e and a is the same as for the rate code output, being ω a conditional probability:

$$e_j = P(\frac{o}{i_j}) I_j \tag{6}$$

Note that I_j (uppercase) is also a probability and not a binary output. In the spiking output model, o is generated according to O, which is obtained by applying to "a" a logistic function $f(a)$ of the type:

$$O = \frac{1}{1 + e^{\frac{-(a-s)}{T}}} \tag{7}$$

where parameter T is usually called temperature and regulates the inclination of the curve. The usual procedure for generating an "o=1" with probability O, is generating a random number between o and 1. If the random number is lower than O a binary "o=1" is produced. Parameter s is the shift of the sigmoid. In the following sections we arbitrarily set T to 0.066 and s to 0.5.

3 Results

At molecular levels, plasticity is related to the amount of Ca^{2+} entering postsynaptic NMDA channels (Artola [1]). When calcium is above a certain threshold, metabolic events lead to synaptic potentiation. Conversely, lower calcium levels yield synaptic depression. Without calcium, neither potentiation nor depression takes place. Notice that calcium entrance depends on the presence of presynaptic action potentials. Therefore, presynaptic action potentials are necessary for homosynaptic plasticity. The need of a presynaptic action potential for allowing homosynaptic plasticity will be called "synaptic directionality".

Several facts related to NMDA channel operation should be taken into account in order to perform a realistic model of plasticity in NMDA channels. We consider two cases: the case in which plasticity is restricted to a single synapse (homosynaptic plasticity) and the case in which there is some influence from other synapses over the one under study (heterosynaptic plasticity).

A0. Homosynaptic plasticity. We will take into account the three following facts:

A1. Synaptic directionality. A presynaptic action potential is necessary to produce any positive or negative change in synaptic plasticity.
A2. Metaplasticity of the LTP threshold. The greater the synaptic weight, the higher the LTP threshold.
A3. Magnesium leakage in NMDA channels: When neurotransmitters are bound in AMPA and NMDA channels, ions like Na^{+} and Mg^{2+} enter through the channels, and produce the postsynaptic voltage. The greater the number of channels, the higher is this postsynaptic voltage. However, once NMDA channels are opened, there is not only an entrance of Ca^{2+} ions, but also an outward leakage of Mg^{2+} and K^{+} ions, that diminishes the expected postsynaptic voltage.

B0. Heterosynaptic plasticity. Two or more nearby synapses are involved, and we consider the following additional facts that are not present in homosynaptic plasticity:

B1. The postsynaptic potential in the synapse under study is incremented by the postsynaptic potential of nearby synapses.
B2. Metaplasticity of the LTD threshold. The higher the weight, the lower the LTD threshold (Mockett et al [3].
B3. The ABS rule.

3.1 Biological Plausibility of Presynaptic Rule Incremental Version

Let us recall the incremental version of the pre-synaptic rule:

$$\triangle \omega = \xi I(O - \omega) \tag{8}$$

Notice that, consistently with the property of synaptic directionality (A.1), if "I" is zero, neither potentiation, nor depression takes place, so term "I" can be regarded as an "allowance term". According to *Methods*, "a" and O can be made proportional in the linear part of the logistic function so that, according to equation 4, I can be replaced by $\frac{O}{\omega}$ or $\frac{a}{\omega}$ in the homosynaptic case, so that:

$$\triangle \omega = \xi(\frac{a}{\omega})(a - \omega) = \xi[(\frac{a^2}{\omega}) - a] \tag{9}$$

which is represented in Fig. 3.a, where the potentiation and depression regions described by Artola et al [1] are present. The curves also exhibit metaplasticity of the LTP threshold.

In the case of heterosynaptic plasticity, the postsynaptic activation a is the sum of the contribution of the synapse being evaluated, ωI, and the potential from nearby synapses, a_n. Recalling property B.1.

$$a = \omega I + a_n \tag{10}$$

Isolating I and taking into account that this term is always positive or zero, the incremental presynaptic rule (Eq. 8) becomes:

$$\triangle \omega = \xi \frac{[a - a_n]_+}{\omega}(a - \omega) \tag{11}$$

where $[a - a_n]_+$ indicates the positive component of $a - a_n$, or zero if $a - a_n$ is negative. Setting a_n to a positive value, as for example 0.3, last expression gives rise to the more realistic curve of figure 3.b in which for low postsynaptic activations ($a < a_n$) neither potentiation nor depression takes place. The curve also exhibits a LTD threshold for $a = a_n$.

This same equation is consistent with the ABS rule in which the horizontal axis represents the presynaptic activity, I, instead of a. Substituting equation 10 in equation 9, equation 9 is expressed in terms of I as:

$$\triangle \omega = \xi I(\omega I + a_n - \omega) \tag{12}$$

which is the curve represented in Fig. 3.c showing a leftward shift of the LTP threshold for greater values of a_n. At the same time, the curve exhibits the uncommon properties described by Artola and Singer [8] in which synaptic potentiation occurs with negligible values of I in the case of very large values of a_n.

Finally, it is possible to take property A.3 into account in the case of heterosynaptic plasticity (see also property B.2).

According to this, the greater the number of AMPA and NMDA channels, that is, the greater the synaptic weight, the greater is also the leakage of K^+ and Mg^{2+} ions through these channels. Representing this leakage by a subtractive term proportional to the weight, the postsynaptic activation is calculated as:

$$a = I\omega - k\omega + a_n \tag{13}$$

In this case the presynaptic rule is expressed in terms of a as:

$$\triangle\,\omega = \xi[\frac{a - a_n}{\omega} + k]_+(a - \omega) \tag{14}$$

yielding fig 3.d, which is consistent with the property of metaplasticity of the LTD threshold.

3.2 Biological Plausibility of the Probabilistic Version of Presynaptic Rule

In previous sections, we simplified the mathematical description of a neuron by making the assumption of linearity in neuron activation function. In this section,

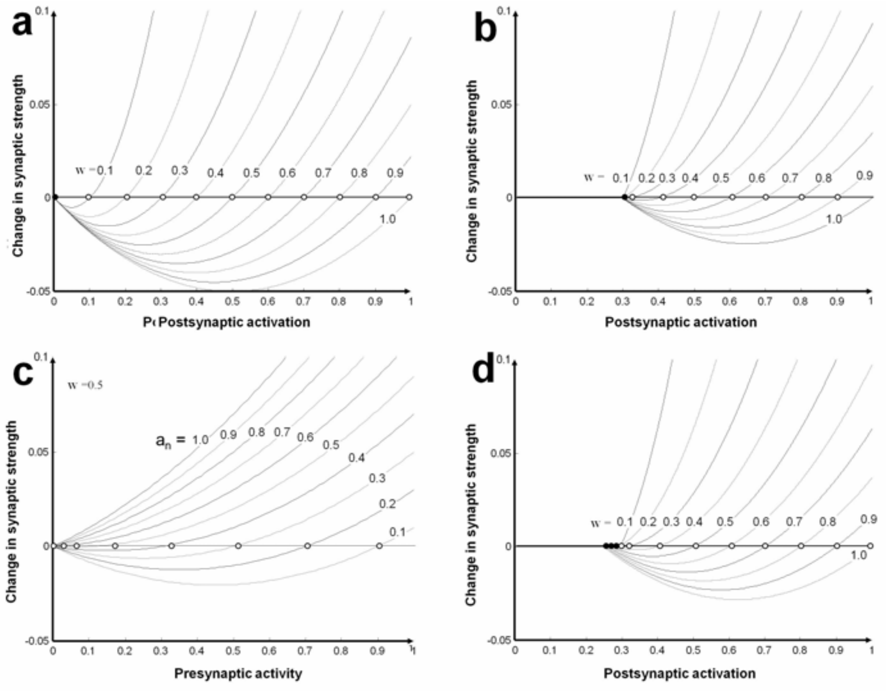


Fig. 3. Metaplasticity and the ABS rule according to the incremental version of presynaptic equation. (a) LTP threshold metaplasticity according to Eq. 9: the different curves are obtained for different values of synaptic weight ω. (b) LTP threshold metaplasticity considering heterosynaptic influence according to Eq. 11. (c) ABS theory according to presynaptic rule incremental version in equation 12. Presynaptic activity is shown in the horizontal axis and the curves are obtained for different values of a_n. The weight was ω = 0.5 (d) Metaplasticity of the LTD threshold according to equation 14.

the neuron output, o, is binary and the probability of a unitary output is given by a non-linear logistic activation function, so that:

$$o = f^*(a) \tag{15}$$

f^* being a function that yields a binary 0 or 1 according to the probability given by O, which is the output given by the sigmoid function f where $O = f(a)$. Here, the variation of synaptic weight $\omega - \omega_0$ is calculated for each activation a and for each initial weight ω_0 by using the probabilistic version of the presynaptic rule: $\omega = P(\frac{o}{i})$. Binary output o is calculated according to Eq. 15 whereas binary input i is calculated from probability I, where in the homosynaptic case, $I = a/w$.

When including a_n (the postsynaptic activation due to other synapses), I is obtained from equation 10 as:

$$I = \frac{[a - a_n]_+}{\omega} \tag{16}$$

Fig. 4.a is the representation of this case. The influence of a_n, that was arbitrarily set to $a_n = 0.3$ in the example, is observed in the rightward shift of 0.3 units of the curves in Figure 4.a.

When property A.3 is taken into account, the presynaptic probability, I, is calculated as:

$$I = [\frac{a - a_n}{\omega} + k]_+ \tag{17}$$

Fig. 4.b corresponds to this case. There is a leftward shift of the LTD threshold with greater values of w (i.e. LTD metaplasticity). Without a_n, LTD metaplasticity is not present. This property is only manifested in the case of heterosynaptic plasticity (property B.2).

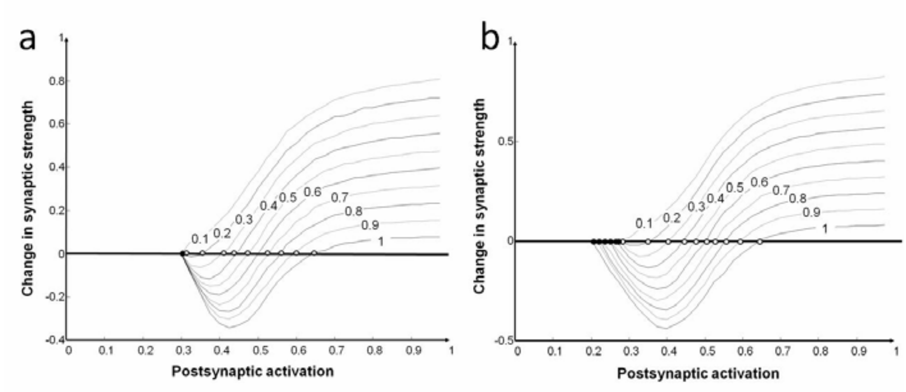


Fig. 4. Plasticity curves calculated according to the probabilistic presynaptic equation using a non-linear activation function (sigmoid). (a) In this case, the curves exhibit LTP threshold metaplasticity considering heterosynaptic influence according to Eq. 16. (b) Metaplasticity of the LTD threshold according to equation 17 and considering property A.3.

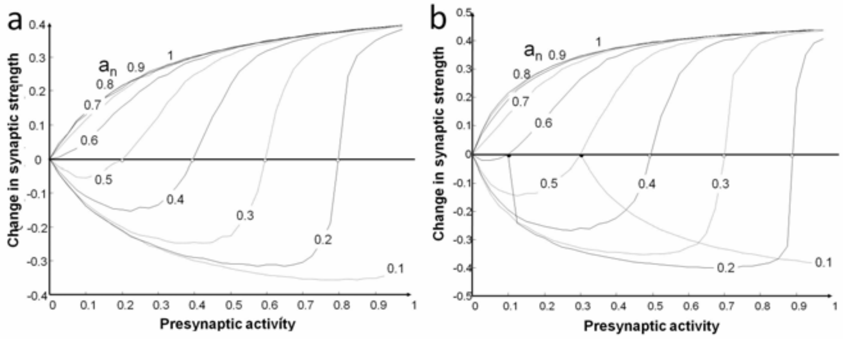


Fig. 5. Synaptic weight variation in terms of presynaptic activity (with the activation of neighbouring synapses an as a parameter). Presynaptic probabilistic equation with the ABS rule (a) Using a probabilistic presynaptic equation with a non-linear activation function (sigmoid).(b) Taken property A.3 into account.

To test whether the conditional probability model of synaptic plasticity is consistent with the ABS rule, the presynaptic probability I is represented in the horizontal axis. To obtain $P(\frac{o}{i})$, o is given by $o = f^*(a)$, a being a function of both I and a_n, according to equation 10. Fig. 5.a seems to encompass more consistently the ABS rule (Fig. 2.b) than our previous attempt in Fig. 3.c. If property A.3 is taken into account (so that a is obtained from Eq. 13 Fig. 5.a become Fig. 5.b, which is even more consistent with the ABS rule (compare with Fig. 2.b).

4 Conclusion

In this work, we have analyzed the presynaptic rule of synaptic plasticity in the light of recent biological findings such as synaptic directionality, metaplasticity of potentiation and depression thresholds, and the influence of nearby synapses (the so called "ABS" rule) over the synapse under study.

The presynaptic rule is consistent with synaptic directionality and LTP metaplasticity, in the case of homosynaptic plasticity, and with LTD metaplasticity and the ABS rule, in the case of heterosynaptic plasticity.

All these properties were studied with: a) the incremental equation of the presynaptic rule, assuming linearity of the activation function around the average activation set-point, and with b) the probabilistic version of presynaptic rule in which a logistic activation function was utilized. While both analyses were consistent with biological findings, the latter seemed to more accurately encompass the qualitative features of experimental curves.

References

1. Artola, A., Brcher, S., Singer, W.: Different voltage-dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex. Nature 347, 69–72 (1990)
2. Abraham, W.C., Bear, M.F.: Metaplasticity: the plasticity of synaptic plasticity. Trends Neurosci. 19, 126–130 (1996)
3. Mockett, B., Coussens, C., Abraham, W.C.: NMDA receptor-mediated metaplasticity during the induction of long-term depression by low-frequency stimulation. European Journal of Neuroscience 15, 1819–1826 (2002)
4. Hebb, D.O.: The organization of behaviour: A neurophysiological theory. John Wiley and Sons, New York (1949)
5. Sejnowski, T.J.: Storing covariance with non linearly interacting neurons. J. Math. Biol. 4, 303–321 (1977)
6. Sejnowski, T.J., Chattarji, S., Stanton, P.K.: Homosynaptic long-term depression in hippocampus and neocortex. Seminars in the Neurosciences 2, 355–363 (1990)
7. Bienenstock, E.L., Cooper, L.N., Munro, P.W.: Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. J. Neurosci. 2(1), 32–48 (1982)
8. Artola, A., Singer, W.: Long-term depression of excitatory synaptic transmission and its relationship to long-term potentiation. Trends Neurosci. 16, 480–487 (1993)
9. Grossberg, S.: Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. Biological Cybernetics 23, 121–134 (1976)
10. Minai, A.A.: Covariance learning of correlated patterns in competitive networks. Neural Computation 9, 667–681 (1997)
11. Abraham, W.C., Tate, W.P.: Metaplasticity: A new vista across the field of synaptic plasticity. Progress in Neurobiology 52, 303–323 (1997)
12. Turrigiano, G.G.: Homeostatic plasticity in neuronal networks: the more things change, the more they stay the same. Trends Neurosci. 22(5), 221–227 (1999)

Dynamics of a Three Neurons Network with Excitatory-Inhibitory Interactions

Carlos Aguirre[1], Juan I. Cano[2], and Eloy Anguiano[2]

[1] GNB, Escuela Politécnica Superior, Universidad Autónoma de Madrid,
28049 Madrid, Spain
Carlos.Aguirre@uam.es

[2] CRL, Escuela Politécnica Superior, Universidad Autonoma de Madrid,
28049 Madrid, Spain
Eloy.Anguiano@uam.es, Inaki.Cano@uam.es

Abstract. Sets of coupled neurons can generate many different patterns in response to modulatory or sensory inputs. The study of how these patterns have been generated from the inputs has been object of great interest in the literature. These studies have been mainly performed by means of computer simulations, based on differential models or phenomenological models. However complete descriptions of the behaviour of sets of coupled neurons are hard to obtain due to the complex behaviour of the dynamics generated even by the simplest neuron models and for the high number of parameters involved. Here we present a study of a three neuron network that appears in models of Central Pattern Generators. The use of a lineal model allows a complete dynamical description of the system, identifying the relevant situations and drawing some conclusions concerning the dynamics of the network.

1 Introduction

Numerical studies of the collective behaviour of ensembles of neurons rely on models of neurons that describe the neuron behaviour on the basis of differential equations of phenomenological models. Differential models (the Hodking-Huxley (HH) model [1], or the Hindmarsh-Rose (HR) model [2]) require high computational effort to reproduce neuronal behaviour such as spiking or bursting. Recently, some models have solved this drawback of the differential models [3,4,5,6]. These new models are built over a phenomenological basis and use iterative two–dimensional maps that present similar neuro–computational properties to that of the differential models such as spiking, bursting or subthreshold oscillations. These models present a low computational effort that makes possible the simulation of big ensembles of coupled neurons during relatively long periods of time, or the exploration of many different regimes determined by inputs, initial conditions and neuron couplings.

Both differential and phenomenological models are adequate for computer simulations, however they are too complex for the complete characterization of the dynamics even for a single neuron, not to say for sets of several coupled

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 90–99, 2011.

neurons. This makes necessary the introduction of simpler models that allow a complete study of the dynamics but preserve the main characteristics of single neuron behaviour. An example of such analytic study is the Kuramoto model [7] developed for the study of many coupled oscillators, similar to the ones present in the peacemaker cells.

A complete study of the dynamics of a three neuron network composed by an excitatory-inhibitory loop that controls a third neuron is presented here. The use of a lineal model allows a complete dynamical description of the system, identifying the relevant situations and drawing some conclusions concerning the dynamics of the model. These network model has been previously studied by means of computer simulations in [8] and [9] as it appears frequently in Central Pattern Generators (CPGs).

2 The Network Model

Here we propose the use of a leaky integrate and fire (IF) iterative model obtained from a differential neuron model presented in eq. 2.1 from [10] . Our model is implemented as an iterated map obtained from a continuous model by an Euler approximation. This approach has been used, for example, in [11] to obtain an iterated map from an continuous neuron model to obtain iterative phenomenological models. This model of neuron is often used in simulation studies. Simple IF neurons have been shown to provide good approximations to the dynamics of more complex model neurons. For example, similar models has been used to explore the different connectivity effects in models of cortical networks [12], in [13] to explore the role of the synaptic background activity in spatio–temporal integration in single pyramidal cells or in [14] to study the chaotic behaviour of a model of cortical network. This simple model will allow us to perform a complete description of the different regimes generated by the dynamics of the network.

We start with the differential model

$$C\frac{dv_i(t)}{dt} = \sum_{j\neq i} I_j(t) + I(t) \tag{1}$$

now we do the approximation $\frac{dv_i}{dt} = \frac{1}{\Delta t}v_i(t+\Delta t) - v_i(t)$. For the simulation of the synaptic current $(I_j(t))$, we model input to neuron i from neuron j at time t as $W_{ij}H(v_j(t) - T_j)$, see eq. 4.19 in [10], where W_{ji} is the coupling strength between neuron j and neuron i. The value W_{ji} is positive if j is an excitatory neuron and negative otherwise. $H(x)$ is the usual Heaviside function ($H(x) = 1$ if $x \geq 0$ and $H(x) = 0$ otherwise) and Tj is the spike threshold of neuron j.

Now, grouping constants , we obtain is the following model:

$$v_i(t+1) = a_i v_i(t) + \sum_{j\neq i} W_{ji}H(v_j(t) - Tj) + W_i I_i(t) \tag{2}$$

where $v_i(t+1)$ means the value of the output of neuron i, v_i, at the next time instant. The value a_i is the descent rate and is required to be positive. We will note

I as the value of the external input, W_{iI} and T_I are respectively the (excitatory) coupling strength and the threshold of the external input. For normalization purposes we ask without loss of generality the condition $a_i + \sum_{i \neq j} |W_{ji}| + W_{iI} = 1$. This condition ensures the bounding of the neuron output in the interval $[-1, 1]$. In this model we consider that neuron j is spiking at time instant t if the value of neuron j, v_j, is over the spike threshold T_j, so a constant value of neuron i over the spike threshold means a tonic spiking, meanwhile an oscillating value around the spike threshold means that the neuron alternates between spiking and resting regimes.

The network object of study is composed by a feedback loop of one excitatory and one inhibitory neuron. The excitatory neuron also receives an external input. This excitatory-inhibitory loop controls the behaviour of a third neuron. Similar circuits have been explored by means on differential models and computer simulations by [11] and [8,9]. The whole network can be seen in figure 1.

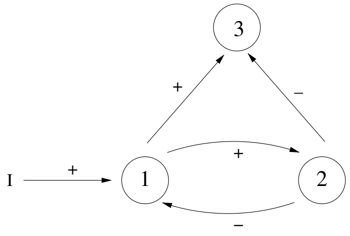


Fig. 1. A network with a feedback loop that controls a third neuron

The dynamical evolution of the network is modelled by a coupled system of piecewise linear iterated maps that vary in time for each neuron, namely

$$\begin{aligned} u_1(t+1) &= a_1 u_1(t) + W_{21} H(u_2(t) - T_2) + W_{I1} H(I - T_I) \\ u_2(t+1) &= a_2 u_2(t) + W_{12} H(u_1(t) - T_1) \\ u_3(t+1) &= a_3 u_3(t) + W_{13} H(u_1(t) - T_1) + W_{23} H(v_2(t) - T_2) \end{aligned} \tag{3}$$

3 Dynamic Behaviour

The feedback circuit is modelled by the coupled maps

$$\begin{aligned} v_1(t+1) &= a_1 v_1(t) + W_{21} H(v_2(t) - T_2) + W_{I1} H(I - T_I) \\ v_2(t+1) &= a_2 v_2(t) + W_{12} H(v_1(t) - T_1) \end{aligned} \tag{4}$$

with $W_{21} < 0$, $W_{I1} > 0$ and $W_{12} > 0$ representing the excitatory-inhibitory loop with the normalization conditions $a_1 - W_{21} + W_{I1} = 1$ and $a_2 + W_{12} = 1$. This

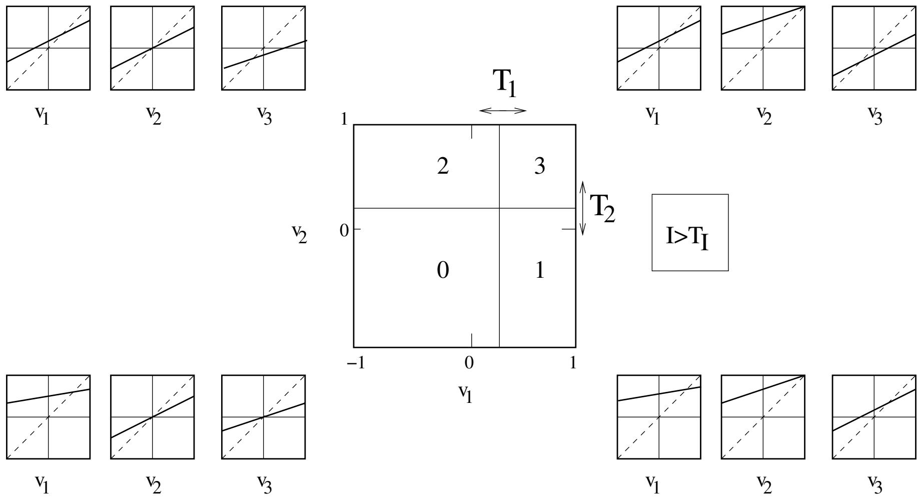


Fig. 2. The v_1, v_2 system for $I > T_I$. The iterated maps operating in each region are depicted near their corners.

dynamical laws are different depending on the values of the thresholds T_1, T_2 and T_I. The thresholds T_1, T_2 divide the square $[-1, 1]$ into four rectangular regions. The equation of the maps changes as the coordinates of the pair v_1, v_2 lies on one of these four regions. This is illustrated in figure 2.

The pair $v_1(t), v_2(t)$ tends to the fixed points corresponding to the maps associated to the region where the point is at a particular moment. Otherwise, the orbit can leave the region on its way to the fixed point, leading to a different dynamical law. The fixed points for each region are depicted in figure 3. In 1 the values of the xed points corresponding to each region are shown. Note that $\frac{W_{21}}{1-a_1} \leq 0$, $\frac{W_{I1}}{1-a_1} \geq 0$ and $\frac{W_{I1}+W_{21}}{1-a_1} \leq \frac{W_{I1}}{1-a_1}$, also from the normalization condition we obtain $\frac{W_{12}}{1-a_2} = 1$.

Examining figure 3 and considering all qualitatively different positions of the thresholds one obtains a complete classification of all possible dynamical behaviours of the system, all the possible situations are listed in table 2.

Table 1. Value of the fixed points

	Region			
	0	1	2	3
$I < T_I$ ○	$(0, 0, 0)$	$(0, 1, \frac{W_{13}}{1-a_3})$	$(\frac{W_{21}}{1-a_1}, 0, \frac{W_{23}}{1-a_3})$	$(\frac{W_{21}}{1-a_1}, 1, \frac{W_{13}+W_{23}}{1-a_3})$
$I > T_I$ ◇	$(\frac{W_{I1}}{1-a_1}, 0, 0)$	$(\frac{W_{I1}}{1-a_1}, 1, \frac{W_{13}}{1-a_3})$	$(\frac{W_{I1}+W_{21}}{1-a_1}, 0, \frac{W_{23}}{1-a_3})$	$(\frac{W_{I1}+W_{21}}{1-a_1}, 1, \frac{W_{13}+W_{23}}{1-a_3})$

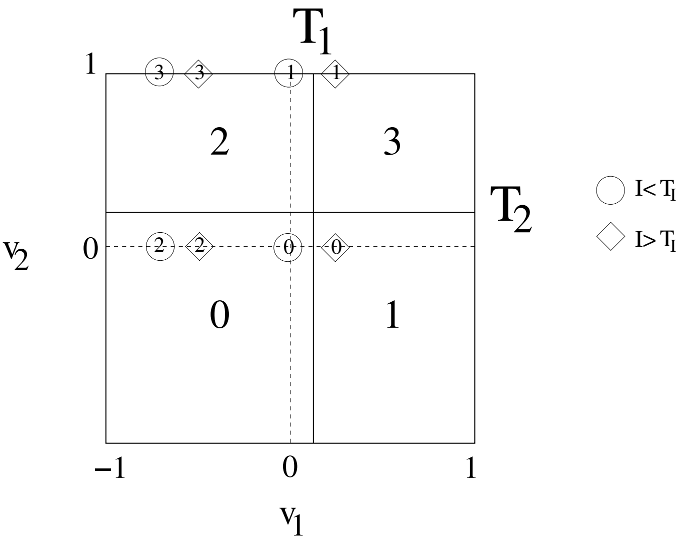


Fig. 3. Dynamical regions and fixed points for the feedback circuit. Numbers in circles represent the fixed point which tend towards the points in the region with the same number when there is no external input. Numbers in diamonds represent the fixed point which tend towards the points in the region with the same number when there is external input.

This way we can further predict the behaviour of our network stating that the two neurons of the excitatory–inhibitory loop will oscillate between spiking and resting periods when the T_1, T_2 pair lies inside the rectangle defined by the four fixed points both in the case of no external input or when we inject an external input to the excitatory neuron. As the output of both neurons cross the respective thresholds in their orbits this produces different rhythmical patterns. This behaviour can be also seen in [8]. Outside of that rectangle the system converges to a fixed point. As expected, high values of the thresholds produce that the corresponding neuron tend to remain resting, meanwhile low thresholds produce tonic bursting.

The injection of external input essentially produces a shift to higher values of the v_1 coordinate of the fixed points. The effect of this shift is that the network can change its regime from resting to oscillations about the threshold or from oscillations to tonic spiking once the external input is applied.

The inhibition role can be seen clearly in the fact that, for a fixed value of the excitatory neuron threshold T_1, this neuron changes his regime from spiking to resting as the inhibitory coupling W_{21} becomes more negative.

The output neuron is fully controlled by the feedback circuit. Its behaviour depends essentially of its spiking threshold value T_3. The neuron will oscillate (but not necessarily about its threshold value) only when the excitatory-inhibitory

Table 2. Behaviour of the excitatory inhibitory network

$I < T_I$				
	Parameters	Convergence	Neuron 1	Neuron 2
$T_2 < 0$	$T_1 < \frac{W_{21}}{1-a_1}$	$(\frac{W_{21}}{1-a_1}, 1, \frac{W_{13}+W_{23}}{1-a_3})$	Spiking	Spiking
	$T_1 > \frac{W_{21}}{1-a_1}$	$(\frac{W_{21}}{1-a_1}, 0, \frac{W_{23}}{1-a_3})$	Resting	Spiking
$T_2 > 0$	$T_1 < \frac{W_{21}}{1-a_1}$	$(\frac{W_{21}}{1-a_1}, 1, \frac{W_{13}+W_{23}}{1-a_3})$	Spiking	Spiking
	$\frac{W_{21}}{1-a_1} < T_1 < 0$	Oscillations		
	$T_1 > 0$	$(0, 0, 0)$	Resting	Resting

$I > T_I$				
	Parameters	Convergence	Neuron 1	Neuron 2
$T_2 < 0$	$T_1 < \frac{W_{I1}+W_{21}}{1-a_1}$	$(\frac{W_{I1}+W_{21}}{1-a_1}, 1, \frac{W_{13}+W_{23}}{1-a_3})$	Spiking	Spiking
	$T_1 > \frac{W_{I1}+W_{21}}{1-a_1}$	$(\frac{W_{I1}+W_{21}}{1-a_1}, 0, \frac{W_{23}}{1-a_3})$	Resting	Spiking
$T_2 > 0$	$T_1 < \frac{W_{I1}+W_{21}}{1-a_1}$	$(\frac{W_{I1}+W_{21}}{1-a_1}, 1, \frac{W_{13}+W_{23}}{1-a_3})$	Spiking	Spiking
	$\frac{W_{I1}+W_{21}}{1-a_1} < T_1 < \frac{W_{I1}}{1-a_1}$	Oscillations		
	$T_1 > \frac{W_{I1}}{1-a_1}$	$(\frac{W_{I1}}{1-a_1}, 0, 0)$	Resting	Resting

loop does. In the other cases, the output value will tend to a fixed point that, again, depending on the value of the threshold T_3, makes the neuron rest or spike tonically.

4 Experimental Results

To support the model proposed before, we developed a small application in Python that simulates the network and tested the model with different values of descent rates, coupling strengths, initial conditions and thresholds that covers most of the behaviours the model can offer[1].

In this section we study three of the simulations results in order to illustrate the values discussed in the previous section. The units used in the graphical representations are arbitrary. All simulations starts with each neuron inactive

[1] The results can be consulted online at: www.ii.uam.es/˜jcano/neuronweb/

Table 3. Neuron parameters for each experiment

Cases	a_1	a_2	a_3	W_{I1}	W_{21}	W_{12}	W_{13}	W_{23}
Case 1	0.4	0.4	0.2	0.4	0.2	0.6	0.6	0.2
Case 2	0.4	0.4	0.2	0.2	0.4	0.6	0.6	0.2
Case 3	0.4	0.4	0.2	0.4	0.2	0.6	0.2	0.6

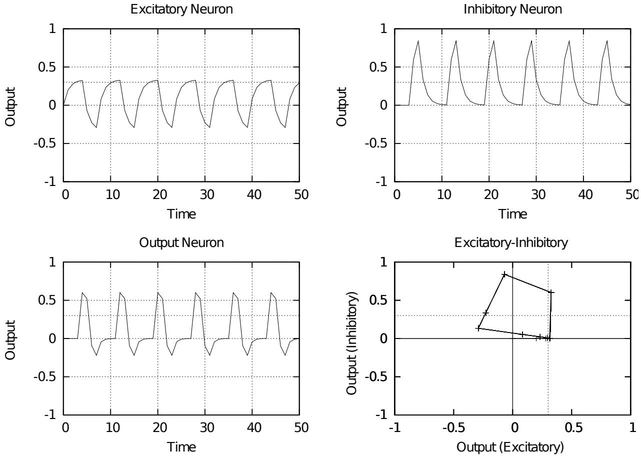


Fig. 4. Representation of the first case

$(v_i = 0), i = 1, 2, 3$, with the input I set at 0.5 and thresholds $T_i,\ i = 1, 2, 3$ set to 0.3. Neuron $N1$ is the excitatory neuron, $N2$ is the inhibitory neuron and $N3$ is the output neuron. The three neurons use similar values, shown on table 3.

Case 1 will represent a spiking model defined when $T_1 < \frac{W_{I1}+W_{21}}{1-a_1}$, this is, oscillations about the spiking threshold. Case 2 is similar to Case 1 with oscillations but no spiking. Case 3 is the case where the network reaches a stable point.

4.1 Case 1

As can be seen in figure 4, the output neuron value graph is oscillatory. Observing the comparative graph (N1 vs N2), the two neurons start at the origin were both neurons are inactive. This is region 0 and the fixed point expected is $(\frac{W_{I1}}{1-a_1}, 0, 0) = (0.33, 0, 0)$. That's the point we reach in the next three steps, on region 1 where N1 is firing.

On this region the fixed point is $(\frac{W_{I1}}{1-a_1}, \frac{W_{12}}{1-a_2}, \frac{W_{13}}{1-a_3}) = (0.33, 1, 0.75)$ so our new point is in that direction. Ascending almost vertically, we go over the N2 threshold and it starts firing, the fixed point is not reached and the direction changes to the fixed point of region 3 $(\frac{W_{I1}+W_{21}}{1-a_1}, \frac{W_{12}}{1-a_2}, \frac{W_{13}+W_{23}}{1-a_3}) = (-0.33, 1, 0.5)$.

Following that direction, N1 stops firing as we enter region 2 with fixed point $(\frac{W_{I1}+W_{21}}{1-a_1}, 0, \frac{W_{23}}{1-a_3}) = (-0.33, 0, -0.25)$. To go under N2 threshold and enter

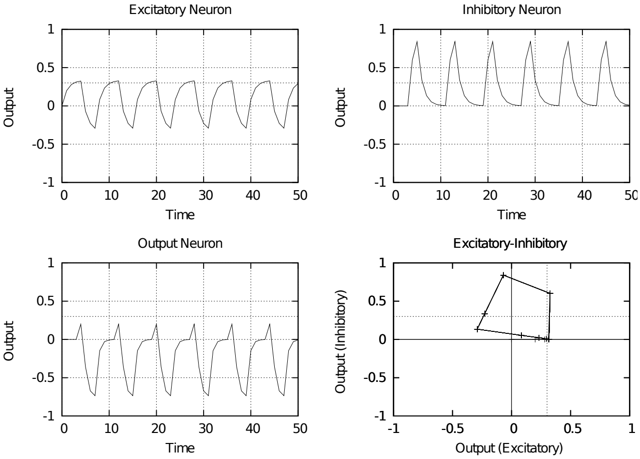


Fig. 5. Representation of the second case

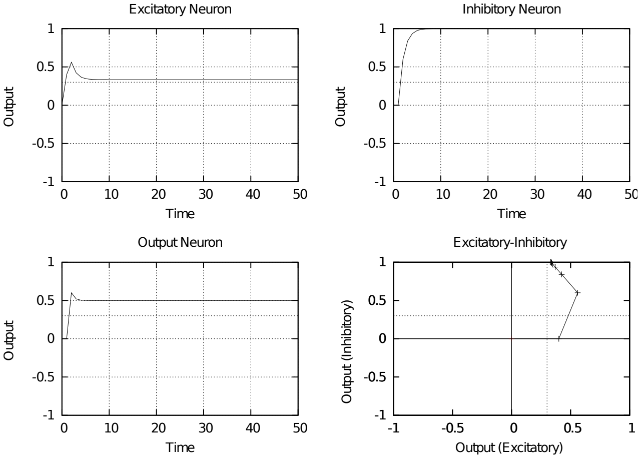


Fig. 6. Representation of the third case

region 0 again we need two steps in that direction and once crossed, we need 3 steps in the direction of the fixed point of region 0, mentioned earlier, to start again at region 1.

4.2 Case 2

This case is similar to the previous one but parameters are selected in such a way that the output neuron does never spike, even when there are oscillations. The plots for this case can be seen on figure 5.

4.3 Case 3

In figure 6 we present a simulation that gets to a stable point. The parameters chosen for this simulation are such that we obtain no oscillations. This means that after some steps the simulation will get to a fixed point on the excitatory-inhibitory loop where the neurons will be firing or resting constantly. The output neuron will be always firing or always resting depending on the value of the threshold.

5 Conclusions

An analytic study of a network consisting of two coupled neurons and an output neuron is presented here. The model used for the neurons is an Euler approximation to the leaky integrate and fire. This model is simple enough to do a complete study of the system dynamic behaviour but sufficiently detailed to present some neuronal features as, for example, the effects of inhibition or the generation of complex rhythm patterns.

This model displays reasonable biological features which may serve, at least, as a toy model for some neural circuits. We concentrated on extracting the functional actions rather than reproducing the biological detail.

Acknowledgments

(CA) is supported by MEC BFU2009-08473. (CA) is partially supported by AYA2009-14212-05. CRL is supported by IBM S.A.E.

References

1. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and application to conduction and excitation in nerve. Journal of Physiology 117, 165–181 (1954)
2. Rose, R.M., Hindmarsh, J.L.: The assembly of ionic currents in a thalamic neuron, I The three dimensional model. Proceedings of The Royal Society of London B 237, 267–288 (1989)
3. Rulkov, N.F.: Modeling of spiking-bursting neural behavior using two–dimensional map. Physical Review E 65, 041922 (2002)
4. Izhikevich, E.M.: Simple Model of Spiking Neurons. IEEE Trans. on Neural Networks 68, 052901 (2003)
5. Shilnikov, A.L., Rulkov, N.F.: Subthreshold oscillations in a map–based neuron model. Physics Letters A 327, 177–184 (2004)
6. Aguirre, C., Campos, D., Pascual, P., Serrano, E.: Neuronal Behavior with Subthreshold Oscillations and Spiking/Bursting Activity Using a Piecewise Linear Two-Dimensional Map. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3696, pp. 103–108. Springer, Heidelberg (2005)
7. Kuramoto, Y.: International Symposium on Mathematical Problems in Theoretical Physics. In: Araki, H. (ed.) Lecture Notes in Physics, vol. 39 (1975)

8. Latorre, R., Rodríguez, F.B., Varona, P.: Characterization of Triphasic Rhythms in Central Pattern Generators (I): Interspike Interval Analysis. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, pp. 160–166. Springer, Heidelberg (2002)
9. Rodríguez, F.B., Latorre, R., Varona, P.: Characterization of Triphasic Rhythms in Central Pattern Generators (II): Burst Information Analysis. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, pp. 167–173. Springer, Heidelberg (2002)
10. Gerstner, W., Kistler, W.M.: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge (August 2002)
11. Izhikevich, E.M.: Neural Excitability, Spiking and Bursting. International Journal of Bifurcation and Chaos 10, 1171–1266 (2000)
12. Aguirre, C., Campos, D., Pascual, P., Serrano, E.: Effects of different connectivity patterns in a model of cortical circuits. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2686, pp. 78–85. Springer, Heidelberg (2003)
13. Bernarder, O., Koch, C., Usher, M.: Synaptic background activity determines spatio–temporal integration in single pyramidal cells. Proceedings of the National Academy of Sciences USA 88, 11569–11573 (1991)
14. van Vreeswijk, C., Sompolinsky, H.: Chaotic Balanced State in a Model of Cortical Circuits. Neural Computation 10, 1321–1372 (1998)
15. Elson, R.C., et al.: Synchronous Behavior of Two Coupled biological Neurons. Physical Review Letters 81(25), 5692 (1998)

The Reversal Potential of Inhibitory Synapses Strongly Impacts the Dynamics of Neural Networks

Santi Chillemi, Michele Barbi, and Angelo Di Garbo

Istituto di Biofisica CNR, Sezione di Pisa,
Via G. Moruzzi 1, 56124 Pisa, Italy
{santi.chillemi,michele.barbi,angelo.digarbo}@pi.ibf.cnr.it
http://www.pi.ibf.cnr.it

Abstract. The balance between inhibition and excitation is at the basis of the maintenance of stable and normal brain electrical activity. Experimental results revealed that inhibitory synapses can become depolarizing as the intracellular concentration of $Cl^{-}1$ of the postsynaptic cells increases. In this work the dynamical behaviour of a network of pyramidal cells coupled to inhibitory Fast-Spiking interneurons was studied by simulations. In particular, in agreement to the experimental data, it was found that the increase of the reversal potential of inhibitory synapses strongly impacts the network dynamics.

1 Introduction

The brain is populated by excitatory neurons and inhibitory interneurons. Among interneurons the most prominent class is that of Fast-Spiking (FS) cells: they are coupled by inhibitory and electrical synapses and are capable of generating synchronous oscillations [1,2]. These cells exert a powerful control of the firing activity of pyramidal neurons by means of somatic and perisomatic synaptic GABAergic contacts [3]. Interneurons receive excitatory inputs from pyramidal cells and then feedback inhibition to them by modulating their firing activities [4]. This mechanism is at the core of the existence of an excitatory-inhibitory feedback loop between pyramidal cells and interneurons [1,2,3,5]. When the cellular mechanisms regulating these processes are compromised (like in epilepsy) the dynamics of this network becomes more complex and not well known. For instance recently it was shown that FS interneurons are the drivers of seizure like phenomena by activating directly pyramidal cells through excitatory GABAergic transmission [6,7]. In fact it was shown that in particular conditions (for instance when an accumulation of Cl^{-} occurs) inhibitory GABAergic synapses can be converted to excitatory. However at present is still unresolved whether interneurons are actively capable of promoting seizure like activity by means of this mechanisms. Motivated by the previous remarks in the present paper we studied the synchronization phenomena occurring in a network of interconnected FS interneurons and pyramidal neurons. To gain realistic results the cells and

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 100–108, 2011.

their synaptic connections were described by biophysical inspired models. In particular, we investigated the conditions affecting and driving the synchronization phenomena in this complex network.

2 Methods

2.1 Model Description

The artificial network is composed by N_{PY} pyramidal neurons and N_{FS} FS interneurons. The pyramidal neuron models are coupled by excitatory synapses and receive inhibitory inputs from the network of FS interneuron. For either the pyramidal neuron or the interneuron, a single compartment biophysical model is employed to describe its spiking activity. In particular, we adopt the pyramidal and interneuron biophysical models proposed in [8]. The mathematical model of the $j-th$ pyramidal neuron reads:

$$C\frac{dV_j}{dt} = I_{P,j} - g_{Na}m_j^3h_j(V_j - V_{Na}) - g_K n_j^4(V_j - V_K) - g_M w_j((V_j - V_M)) \\ -g_L(V_j - V_L) + I_{PP,j} + I_{IP,j} + \eta_P \xi_{P,j}(t) \quad (1)$$

$$\frac{dm_j}{dt} = \alpha_{m,j}(1 - m_j) - \beta_{m,j}m_j \quad (2)$$

$$\frac{dh_j}{dt} = \alpha_{h,j}(1 - h_j) - \beta_{h,j}h_j, \quad (3)$$

$$\frac{dn_j}{dt} = \alpha_{n,j}(1 - n_j) - \beta_{n,j}n_j, \quad (4)$$

$$\frac{dw_j}{dt} = \frac{w_{j,\infty} - w_j}{\tau_{j,w}}, \quad (5)$$

where $C = 1$ $\mu F/cm^2$, $I_{P,j} = I_P\,(j = 1, 2, ..N)$ is the external stimulation t. The maximal specific conductances and the reversal potentials are respectively: $g_{Na} = 100$ mS/cm^2, $g_K = 80$ mS/cm^2, $g_M = 1$ mS/cm^2, $g_L = 0.15$ mS/cm^2 and $V_{Na} = 50$ mV, $V_K =$ -100 mV, $V_M =$ -100 mV, $V_L =$ - 72 mV. The rate variables describing the currents are defined by: $\alpha_{m,j}(V_j) = 0.32(V_j{+}54)/[1{-}exp((V_j{+}54)/4)]$,$\beta_{m,j}(V_j) = 0.28(V_j{+}27)/[exp((V_j{+}27)/5){-}1]$, $\alpha_{h,j}(V_j) = 0.128exp(-(V_j + 50)/18)$, $\beta_{h,j}(V_j) = 4/[1 + exp(-(V_j + 27)/5)$, $\alpha_{n,j}(V_j) = 0.032(V_j + 52)/[1 - exp(-(V_j + 52)/5)]$, $\beta_{n,j}(V_j) = 0.5exp(-(V_j + 57)/40)$, $w_{j,\infty} = 1/[1 + exp(-(V_j + 35)/10)]$, $\tau_{j,w} = 400/[3.3exp((V_j + 35)/20) + exp(-(V_j + 35)/20)]$. In this model the onset of periodic firing occurs through an Hopf bifurcation for $I_P \cong 3.25$ $\mu A/cm^2$ with a well defined frequency ($\nu \cong 5Hz$).

The current $I_{PP,j}$ arises from the excitatory coupling of the $j-th$ pyramidal neuron with the other cells, while $I_{IP,j}$ describes the inhibitory current due to the coupling with the network of interneurons. These currents will be defined in the next section. To reproduce the membrane potential fluctuations each $j-th$ cell model is injected with the noisy current $\eta_P \xi_{P,j}(t)$, $\xi_{P,j}$ being an uncorrelated Gaussian random variable of zero mean and unit standard deviation $< \xi_{P,i}, \xi_{P,j} >= \delta_{ij}, i \neq j = 1, 2, 3, , N_{PY})$. The adopted value of the parameter η_P was chosen to have a realistic amplitude of the fluctuations of membrane potential.

The biophysical mathematical model of the $j-th$ FS interneuron reads:

$$C\frac{dV_j}{dt} = I_{F,j} - g_{Na}m_j^3h_j(V_j - V_{Na}) - g_K n_j^4(V_j - V_K) - g_L(V_j - V_L) \\ +I_{FF,j} + J_{FF,j} + I_{PF,j} + \eta_F \xi_{F,j}(t) \qquad (6)$$

$$\frac{dm_j}{dt} = \alpha_{m,j}(1 - m_j) - \beta_{m,j}m_j \qquad (7)$$

$$\frac{dh_j}{dt} = \alpha_{h,j}(1 - h_j) - \beta_{h,j}h_j, \qquad (8)$$

$$\frac{dn_j}{dt} = \alpha_{n,j}(1 - n_j) - \beta_{n,j}n_j, \qquad (9)$$

where $C = 1\ \mu F/cm^2$, $I_{F,j} = I_F\ (j = 1, 2, ..N)$ is the external stimulation current. The maximal specific conductances,the reversal potentials and the rate variables are equal to those adopted for the pyramidal cell model. In this model the onset of periodic firing occurs through an Hopf bifurcation for $I_F \cong 1.04$ $\mu A/cm^2$ with a well defined frequency $(\nu \cong 2Hz)$.

The current $I_{FF,j}$ arises from the inhibitory coupling of the $j-th$ FS interneuron with the other cells, while $J_{FF,j}$ describes the current due to the electrical coupling (gap-junction) among interneurons; lastly $I_{PF,j}$ describes the excitatory current due to the coupling with the network of pyramidal neurons. These currents will be defined in the next section. To reproduce the membrane potential fluctuations each $j-th$ cell model is injected with the noisy current $\eta_F \xi_{F,j}(t)$, $\xi_{F,j}$ being an uncorrelated Gaussian random variable of zero mean and unit standard deviation $< \xi_{F,i}, \xi_{F,j} >= \delta_{ij}, i \neq j = 1, 2, 3, , N_{FS})$ and $< \xi_{P,i}, \xi_{F,j} >= 0$. The value of the η_F was chosen in order to get realistic amplitude of the fluctuation of membrane potential.

The reason of using a single compartment model of each cell is motivated by computational constraints. The simulation will be performed by using up to 100 coupled neuron models, and this requires a high computational cost. Therefore, for the aim of the present work, the choice of using a single compartment biophysical model of each cell is a good compromise between two requirements: computational advantages and realistic network of coupled neurons.

2.2 Synaptic Coupling

The excitatory synaptic coupling among pyramidal cells is assumed to be all-to-all. The excitatory synaptic current acting on the $j-th$ pyramidal cell is defined by

$$I_{PP,j} = -\frac{1}{N_{PY}-1}\sum_{k\neq j} g_e s_{PP,k}(t)(V_j - V_{PP}) \tag{10}$$

where $g_e = 0.5mS/cm^2$ represents the maximal amplitude of the excitatory coupling, the function $s_{PP,k}(t)$ describes the time evolution of the postsynaptic current and V_{PP} is the corresponding reversal potential. According to [8] the time evolution of $s_{PP,k}(t)$ is described by

$$\frac{ds_{PP,k}(t)}{dt} = T(V_k)(1 - s_{PP,k}) - s_{PP,k}/\tau_e \tag{11}$$

where $T(V_k) = 5(1 + tanh(V_k/4)$ and $\tau_e = 2ms$ is the decay time constant.

Similarly the inhibitory synaptic coupling among FS interneurons is assumed to be all-to-all and the synaptic current on the $j-th$ interneuron reads

$$I_{FF,j} = -\frac{1}{N_{FS}-1}\sum_{k\neq j} g_i s_{FF,k}(t)(V_j - V_{FF}) \tag{12}$$

where $g_i = 0.25mS/cm^2$ represents the maximal amplitude of the inhibitory coupling and V_{FF} is the corresponding reversal potential. The time evolution of $s_{FF,k}(t)$ is described by

$$\frac{ds_{FF,k}(t)}{dt} = T(V_k)(1 - s_{FF,k}) - s_{FF,k}/\tau_i \tag{13}$$

where $T(V_k) = 2(1 + tanh(V_k/4)$ and $\tau_1 = 10ms$ is the decay time constant.

The pyramidal cells excite the network of FS cells and the corresponding excitatory current acting on the $j-th$ interneuron is defined as

$$I_{PF,j} = -\frac{1}{N_{PY}-1}\sum_{k\neq j} g_{PF} s_{PF,k}(t)(V_j - V_{PF}) \tag{14}$$

where g_{PF} represents the maximal amplitude of the excitatory coupling and $V_{PF} = V_{PP}$ is the corresponding reversal potential. The time evolution of $s_{PF,k}(t)$ is driven by

$$\frac{ds_{PF,k}(t)}{dt} = T(V_k)(1 - s_{PF,k}) - s_{PF,k}/\tau_e \tag{15}$$

where $T(V_k) = 5(1 + tanh(V_k/4)$ and $\tau_e = 2ms$ is the decay time constant.

The network of FS interneurons feedback inhibition to the pyramidal neurons and the inhibitory current of the $j-th$ cell is given by

$$I_{FP,j} = -\frac{1}{N_{FS}-1}\sum_{k\neq j} g_{FP} s_{FP,k}(t)(V_j - V_{FP}) \tag{16}$$

where g_{FP} represents the maximal amplitude of the inhibitory coupling and $V_{FP} = V_{FF}$ is the corresponding reversal potential. The time evolution of $s_{FP,k}(t)$ is determined by

$$\frac{ds_{FP,k}(t)}{dt} = T(V_k)(1 - s_{FP,k}) - s_{FP,k}/\tau_i \qquad (17)$$

where $T(V_k) = 2(1 + tanh(V_k/4)$ and $\tau_i = 10ms$ is the decay time constant.

The electrical coupling among FS interneurons is all-to-all and the corresponding current on the $j - th$ cell is defined as

$$J_{FF,j} = \frac{1}{N_{FS} - 1} \sum_{k \neq j} g_{el}(V_j - V_k) \qquad (18)$$

where g_{el} is the coupling amplitude. The parameters values g_i, g_e, g_{PF}, g_{FP} are those adopted in [8].

3 Results

3.1 Effects of Couplings on the Networks Dynamics

Let us start by investigating the synchronization properties of pyramidal neurons and FS interneurons when the coupling between the two networks is set off. The corresponding results are reported in figure 1. The left panel shows the results obtained by setting off the excitatory synapses among pyramidal neurons and the inhibitory synapses among FS interneurons. In the middle panel are plotted the results showing as the introduction of coupling modifies the dynamics of both networks. The excitatory synapses promotes synchronous firing of pyramidal cells, while the inhibitory coupling among FS interneurons leads to the decrease of their firing activity.

For the adopted parameter values the increase of the conductance of the inhibitory synapses does not significantly improve the synchronization level between FS cells (data not shown). Instead the increase of the amplitude of the stimulation current I_F of the interneurons leads to a more synchronous network activity (data not shown). In the right panel of figure 1 are reported the results obtained with an higher value of the conductance (g_e) of the excitatory synapses. In this case the network of pyramidal cells exhibits a bursting regime. Further increase of the value of g_e determines an enlargement of time windows where the bursting regime occurs (data not shown). The data on the right panel of figure 1 also show that the presence of the electrical synapses among FS interneurons promotes an increase of the synchrony of their firing activities.

Let us now start the discussion of the results describing the dynamics of both networks when the coupling between them is set on. The results reported in the left panel of figure 2 show that the exitatory synapses (the inhibitory synapses from FS interneurons to pyramidal neurons are set off) from the pyramidal cells to the FS interneurons promote a bursting regime in this network (see for comparison the middle panel of figure 1). In such way the level of synchronous

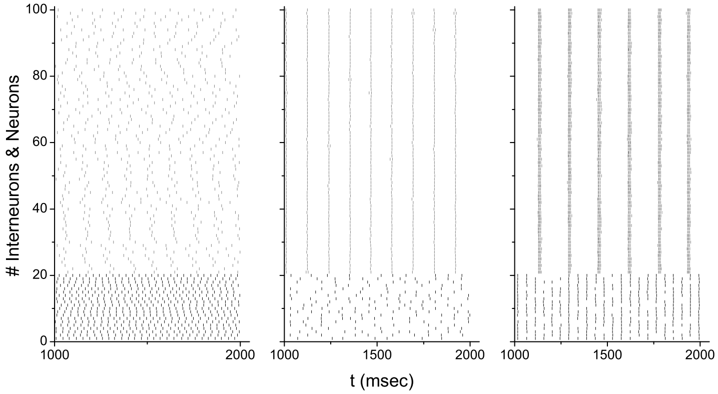


Fig. 1. Synchronization properties of pyramidal neurons and FS interneurons in absence of coupling between the two networks ($g_{FP} = g_{PF} = 0$). Left panel: $g_e = 0$, $g_i = 0$, $g_{el} = 0$, $I_P = 3.5\mu A/cm^2$, $I_F = 1.3\mu A/cm^2$, $N_{PY} = 80$, $N_{FS} = 20$. Middle panel: as the left panel except for $g_e = 0.2mS/cm^2$, $g_i = 0.25mS/cm^2$. Right panel: as the left panel except for $g_e = 0.5mS/cm^2$, $g_i = 0.25mS/cm^2$, $g_{el} = 0.1mS/cm^2$. The noise amplitudes are $\eta_P = \eta_F = 0.2\mu A/cm^2$. For all panels the gray vertical bars represent the firing times of the pyramidal cells, while the black ones those of the FS interneurons.

discharges of the inhibitory cells increases. The results plotted on the middle panel of figure 2 show that not appreciable change on the firing activities of pyramidal cells occurs when the inhibitory synapses from FS interneurons to the excitatory network are set on (see for comparison the results reported in the right panel of figure 1). When it is $g_{FP} \neq 0$ and $g_{PF} \neq 0$ the bursting activity in the network of excitatory cells in almost abolished and the cells fire nearly synchronously. The effect of the reciprocal coupling between the two networks leads to a little increase of the synchrony levels among inhibitory cells. Moreover there is a partial phase locking regime between excitatory and inhibitory cells and this behavior arises as a consequence of the reciprocal coupling between the two networks.

3.2 Reversal Potential of the Inhibitory Synapses and Networks Dynamics

The results of several experimental investigations have shown that seizure afterdischarges induced by electrical stimulation promotes the accumulation of intracellular Cl^-. In these conditions inhibitory GABAergic synapses on pyramidal cells exert a depolarizing action. This phenomenon also occurs among FS interneurons: i.e. the inhibitory synapses connecting the cells of this network become depolarizing [6,7]. This phenomenon is driven by the reversal potential value of GABAergic synapses [7]. In our computational approach we studied how the change of the reversal potential value of GABAergic synapses affects the dynamics of the pyramidal and FS cells networks.

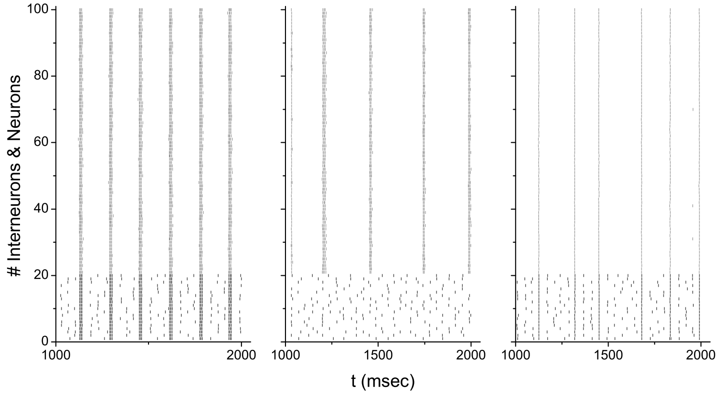


Fig. 2. Effects of coupling betweeen pyramidal neurons and FS interneurons on the networks dynamics. Left panel: $g_{PF} = 0.3mS/cm^2$, $g_{FP} = 0$. Middle panel: $g_{PF} = 0$, $g_{FP} = 1mS/cm^2$. Right panel: $g_{PF} = 0.3mS/cm^2$, $g_{FP} = 1mS/cm^2$. For all panels it is: $g_e = 0.5$, $g_i = 0.25$, $g_{el} = 0$, $I_P = 3.5\mu A/cm^2$, $I_F = 1.3\mu A/cm^2$, $N_{PY} = 80$, $N_{FS} = 20$. For all panels the gray vertical bars represent the firing times of the pyramidal cells, while the black ones those of the FS interneurons.

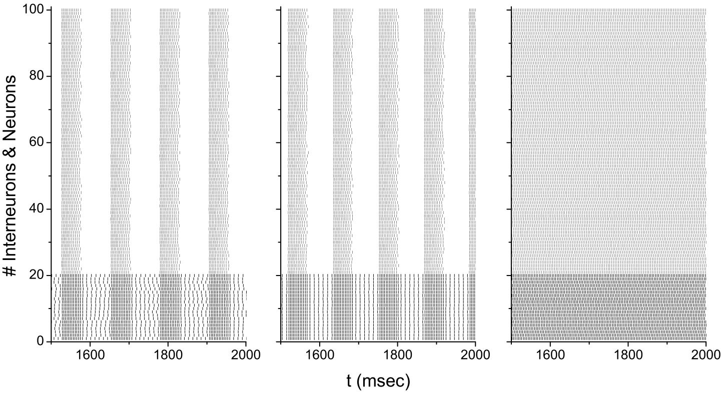


Fig. 3. Effects on the neural networks dynamics arising from the change of the reversal potential value of the inhibitory synapses. For all panels it is: $g_e = 0.5$, $g_i = 0.25$, $g_{PF} = 0.3mS/cm^2$, $g_{FP} = 1mS/cm^2$, $I_P = 3.5\mu A/cm^2$, $I_F = 1.3\mu A/cm^2$, $N_{PY} = 80$, $N_{FS} = 20$. Left panel: $V_{FP} = V_{FF} = -50$, $g_{el} = 0$. Middle panel: $V_{FP} = V_{FF} = -50$, $g_{el} = 0.1$. Right panel: $V_{FP} = V_{FF} = -45$, $g_{el} = 0$.

In the left panel of figure 3 the results obtained in the case $V_{FP} = V_{FF} = -50mV$ are shown. The comparison with the results corresponding to $V_{FP} = V_{FF} = -80$ (see right panel of figure 2) shows that the prominent effect is a strong increase of the firing activity of both networks. The activity of each network is characterized by bursting and that occurring in the population of coupled FS cells is caused by the presence of excitatory synapses from pyramidal neurons. Then, FS interneurons feed forward depolarizing inputs to pyramidal neurons thanks to the higher value of the reversal potential. In this way the interaction of the two networks is characterized by a positive feedback loop. In the middle panel of figure 3 the results obtained in the presence of gap-junctions among FS interneurons are plotted. The firing activities of both networks does not exhibit new features with respect to the previous case. However a more detailed inspection shows that the presence of gap-junctions promotes a more synchronous firing of the cells of both networks.

From general considerations it is expected that a dynamical systems driven by a positive feedback loop can exhibit instability, depending on the parameter values. This behaviour arises also in the system of coupled neural networks that is investigated here. In fact a little increase of the value of the reversal potential value of the inhibitory synapses leads to the dynamical behaviour shown in the right panel of figure 3. This firing pattern can be interpreted as the emergence of a seizure-like phenomenon.

4 Conclusions

Inhibitory interneurons exert a powerful control of the firing activities of pyramidal cells [1,2,3,4,5]. However recent experimental results pointed out that in particular conditions (accumulation of intracellular Cl^-) the inhibitory synapses can depolarize the postsynaptic cells [7]. This phenomenon occurs because the reversal potential value of the inhibitory currents increases. Motivated by these experimental data, we studied the dynamical behaviour of a biophysical inspired network of two coupled populations of cells: the first population is composed by coupled FS interneurons (coupled by inhibitory and electrical synapses), while the second one is constituted by coupled pyramidal cells (coupled by excitatory synapses). In agreement to the experimental model proposed in [6,7] the two networks are connected by excitatory synapses (from pyramidal cells to FS interneurons) and inhibitory synapses (from FS interneurons to pyramidal neurons).

When the coupling between the two networks is set off, the excitatory synapses promote synchronous firing of pyramidal cells, while the inhibitory synapses among FS interneurons lead to the decrease of their firing activity. Increasing the conductance of the excitatory synapses determines the emergence of bursting in the network of pyramidal cells. Moreover, it was found that the presence of the electrical synapses among FS cells promotes the increase of their firing synchrony. When the coupling between the two networks is set on the dynamical behaviour of the two neural populations become richer. In particular bursting regimes also occur in the network of FS cells. Overall, the reciprocal coupling between the two networks promotes the increase of the synchrony among inhibitory cells. Moreover incomplete phase locking states between excitatory and inhibitory cells were also observed. Next we investigated how the value of the reversal potential of the inhibitory synapses affects the dynamical behaviour of both networks. It was found that the increase of the value of the reversal potential leads to the increase of the firing activity of both networks (bursting firing activities). Finally, as the inhibitory inputs become depolarizing a positive feedback loop between the population of pyramidal cells and the population of FS interneuron exists. Then, it was shown that as the value of the reversal potential of the inhibitory synapses overcome a give threshold, a transition toward seizure-like dynamics occurs (see figure 3).

References

1. Freund, T.F.: Interneuron diversity series: Rhythm and mood perisomatic inhibition. Trends Neurosci. 26, 489–495 (2003)
2. Fries, P.: A mechanism for cognitive dynamics: neuronal communication through neuronal coherence. Trends Cogn. Sci. 9, 474–480 (2005)
3. Mann, E.O., Paulsen, O.: Role of GABAergic inhibition in hippocampal network oscillations. Trends Cogn. Sci. 30, 343–349 (2007)
4. Fries, P., Nikolik, D., Singer, W.: The gamma cycle. Trends Neurosci. 30, 309–316 (2007)
5. Hestrin, S., Galarreta, M.: Electrical synapses define networks of neocortical GABAergic neurons. Trends Neurosci. 28, 304–309 (2005)
6. Fujiwara-Tsukamoto, Y., Isomura, Y., Imanishi, M., Fukai, T., Takada, M.: Distinct types of ionic modulation of GABA actions in pyramidal cells and interneurons during electrical induction of hippocampal seizure-like network activity. Eur. J. Neurosci. 25, 2713–2725 (2007)
7. Fujiwara-Tsukamoto, Y., Isomura, Y., Imanishi, M., Ninomiya, T., Tsukada, M., Yanagawa, Y., Fukai, T., Takada, M.: Prototypic Seizure Activity Driven by Mature Hippocampal Fast-Spiking Interneurons. J. Neurosci. 30, 13679–13689 (2010)
8. Olufsen, M., Whittington, M., Camperi, M., Kopell, N.: New roles for the gamma rhythm: population tuning and preprocessing for the beta rhythm. J. Comp. Neurosci. 14, 33–54 (2003)

Doman's Inclined Floor Method for Early Motor Organization Simulated with a Four Neurons Robot

Francisco Javier Ropero Peláez and Lucas Galdiano Ribeiro Santana

Center for Mathematics, Computation and Cognition; Federal University of ABC
Avenida dos Estados 5001, Santo André, Brazil
{fjavier.ropero,galdianosantana}@gmail.com
http://www.ufabc.edu.br/

Abstract. Early motor organization using the Domans inclined floor method is simulated with a four neurons robot. A LEGO robot controlled by a biologically plausible neural network performs the same kind of "inclined floor" training that is given by parents to young babies for early motor organization. When the inclined floor training is applied to the robot, it organizes its motor behavior in a manner that is analogous to the motor organization seen in babies. In this way the simulation with the robot could help to understand the kind of neural processes that are involved in early motor organization.

Keywords: Doman's method, inclined floor, neurological reorganization, neural networks, intrinsic plasticity, synaptic plasticity.

1 Introduction

Temple Fay (1895-1963) was a famous neurosurgeon who developed well established surgery techniques like the tentorial section technique for large acoustic neuromas, the fluid restriction technique in head trauma and refrigeration techniques for the control of cancer. Working with physiotherapist Glenn Doman, he led a team of medical professionals who developed several techniques for improving the intellectual and motor skills of normal and brain-injured children. They founded "The Institutes for the Achievement of Human Potential" in Pennsylvania.

Some of the techniques developed by the institutes (see www.iahp.org) appear in many videos that were recorded and placed on the internet by parents all over the world. Regarding intellectual skills, these videos show how less than one-year infants are capable of reading and even performing mathematical operations such as addition and multiplication.

Regarding motor skills, these videos also show how infants start to crawl with less than two months old when they are stimulated through the use of the "inclined floor" technique.

The inclined floor stimulation technique[3] consists of placing a child in a prone position over an inclined floor or ramp, looking to the lower end of the

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 109–118, 2011.

ramp. The ramp should be slippery and inclined in such an angle that when the child moves forward, he/she slips and falls a little over the ramp. The lateral walls of the ramp are painted in black and white stripes[4]. (Doman et al. 2006) Parents are encouraged to cheerfully make sounds like clapping their hands when the baby shifts forward over the ramp. Glenn Doman describes the foundations of this method:

"When a well infant is put on the floor on a belly-down position he does not know how to crawl... and his arms and legs moves freely... these movements will occasionally result in pushing him forward. When this has happened by accident over and over again, the baby discovers which movements have that effect and which do not. He then discovers how the movements that push him forward feel as distinguished from how the movements that do not push him forward feel. He finally discovers how to purposely reproduce those motions that push him forward and how to synchronize them with each other into a pattern."[3]

Temple Fay and Glenn Doman believe that the nervous system is a sophisticated cybernetic system in exactly the terms Norbert Wiener defined it in his book Cybernetics.[11]

"It was clear that the human brain, like Dr. Wieners self-regulating system, operated like a cybernetic loop. In human beings this loop, which begins in the environment, follows sensory pathways to the brain and motor pathways from the brain back to the environment."[3]

It was this idea that stimulated us to construct a simple cybernetic system in the form of a little robot (see Fig. 1) with a rudimentary sensory system that consisted in an acceleration detector (mimicking the vestibular system), a microphone (for mimicking audition) and a light detector sensor (as a rudimentary visual system). The information from this sensory system arrives to a four neurons neural-network that could be seen as the robot's nervous system. Each of these neurons is interconnected to each other and also to two servomotors. The motor pathway of our rudimentary cybernetic system is represented by the outputs of the neurons that command the two servomotors. Each neuron activates either a clock or anti-clock wise direction of each motor.

The environment in our cybernetic system is represented by a ramp, by the black and white stripes painted on the ramp and by the sound produced by a baby shaker.

For describing the cybernetic loop that controls the dynamic of the robot we will paraphrase the above Glenn Doman's explanation:

"When the robot is put on the ramp it is not initially predisposed to move in any specific direction and its wheels responds to its neuron's commands by moving randomly, occasionally resulting in pushing the robot down the ramp. This part of the system, from neurons to wheels, corresponds to the feed forward part of the robot's cybernetic loop. When the robot's descension happened by accident over and over again, the robot 'discovers' how the movements that push him forward feel as distinguished from how the movements that do not push him forward feel. The difference between these two kinds of movements is in the size of the sensory inputs because, when descending, the robot slips

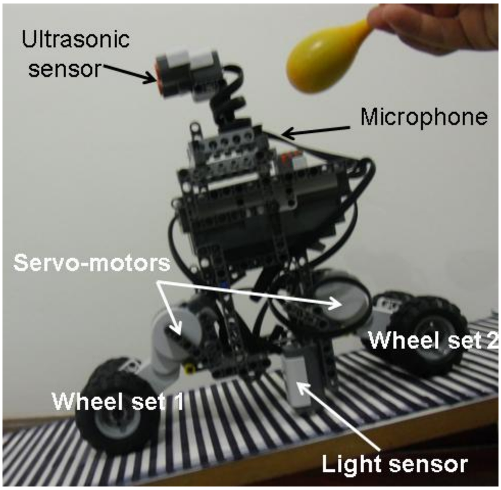


Fig. 1. The Robot

a little aided by gravity; this fact produces a greater downward acceleration (perceived by the acceleration detector). When the robot descends, the greater downward acceleration also makes the black stripes alternate quicker in front of the light detector than when the robot ascends. Another kind of stimulus that appears when the robot moves downwards is the sound of the baby-shaker. This sound stimulation tries to mimic the encouraging words of parents when the baby performs downward movements over the ramp. This part of the system, from environment to neurons, corresponds to the feedback part of the robot's cybernetic loop.

The robot finally discovers how to purposely reproduce those motions that push him downwards and how to synchronize them with each other into a pattern. This happens, as will be shown, because a successful downward movement of the robot is immediately followed by an intense rewarding stimulation from sensory inputs to the neurons that contributed to the movement, allowing enhanced plasticity processes between these neurons."

A similar kind of process has been detected in real organisms. In the case of the vestibular system, Takeda and collaborators[10] discovered that vestibular stimulation enhances acetylcholine production in the rat hippocampus (a brain structure involved in short-term memory). Related to this, Manabe and collaborators research[7], shows that acetylcholine is a neuro-modulator that contributes to synaptic weights modulation or, in other words, to plasticity processes between neurons.

At the light of the above processes, it would not be very audacious to hypothesize that when a child is experiencing body acceleration (i.e. vestibular stimulation) due to inclined floor therapy, plasticity processes in his hippocampus could help him to consolidate short term memories related to the motor tasks he is performing (as for example learning how to crawl).

This paper is organized as follows. In section two we present the robotic system and the neural architecture that allows the robot to learn. We also explain how neurons were modeled highlighting the cybernetic principles that are implicit in the equations. In section 3 we place the robot on the inclined floor, and evaluate its behavior with different kind of stimuli.

In section 4 we study the practical consequences that could be derived from the tests with the robot for improving the performance of the inclined floor technique.

2 Architecture of the Robot

2.1 Architecture and Connectivity of the Robotic System

The robotic structure was built with a LEGO MindstormsNXT toolkit. The software for modeling the neural network and the sensory and motor commands were written in Matlab using the RWTH Mindstorms NXT toolbox which is of public domain and available on the internet.

The feed-forward path of the robot consisted in a set of four fully interconnected neurons (see Fig. 2), each one of them controlling a specific clockwise or anticlockwise movement of two servomotors acting over two wheel sets. The feed-back path of the robot consisted in three sensors: a microphone, a light sensor and an ultrasonic sensor. The contribution of these sensors is summed and this sum increases the activation of the four neurons.

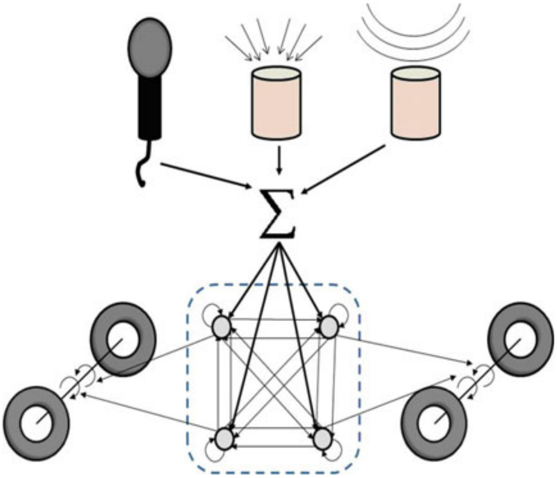


Fig. 2. Organization of the robotic system. Three types of sensors send their information to the four-neurons robot "brain". Each neuron commands a specific clock or anticlockwise movement of each of the two wheel sets.

2.2 Neural Organization of the Robot

As previously mentioned, the robot is controlled by the activity of a four-neurons' neural-network. This fully interconnected network is composed of four biologically plausible excitatory-rate code neurons. Only the connection between neurons has modifiable synapses (weights). Recurrent synapses (connected to the same neuron) are not modifiable. We arbitrarily assigned a value of 0.7 to the weights of these synapses. The effect of recurrent connections is that the neurons' activation decays over time as it happens with cortical pyramidal neurons.

It is remarkable that Temple Fay envisioned that cybernetic principles were important to understand the central nervous system. In fact these principles are at the heart of neuronal functioning and are determinant for allowing learning in our robotic structure. These principles act in the modulation of synaptic weights (synaptic plasticity) and in the adjustment of the neurons firing interval (intrinsic plasticity).

Synaptic Plasticity. Synaptic plasticity refers to the modulation of the efficacy of information transmission between neurons, being related to the regulation of the number of ionic channels in synapses.

The first model of synaptic plasticity was postulated by Hebb and is known as the Hebb rule, that may be stated as follows: when two neurons fire together they wire together or, in other words, the synaptic strength between neurons with correlated firing tends to increase[6]. Mathematically the change in the synaptic strength (synaptic weight) between neurons$i$ and j is calculated by the product of the output of neuron I, O_i , and the input I_j (which corresponds to the output of neuron j) multiplied by a learning constant.

$$\triangle\, \omega_{ij} = \varepsilon O_i I_j \tag{1}$$

Despite its importance, this formulation has a limitation: because in the equation all variables are positive, it leads synaptic weights to grow without bound. For allowing synaptic depression, alternative equations that take into account more recent biological studies have been formulated[5]. The equation that was adopted for our simulation of synaptic plasticity, due to its biological plausibility, is Grossberg's presynaptic learning rule which is as follows:

$$\triangle\, \omega_{ij} = \varepsilon I_j (\alpha_i - \omega_{ij}) \tag{2}$$

Where α is the activation given by the sum of synaptic contributions. In this equation, the subtractive term is equivalent to the negative feedback of a cybernetic loop. This negative feedback allows the artificial neuron to exhibit metaplasticity, a very important characteristic of biological neurons[2][8]. Metaplasticity slows down the process of weight increment or decrement, making it more difficult for the neuron to become either saturated or inactive. In the case of the equation, it is easy to understand that the increment of weight is less for big initial weights than for smaller weights.

Metaplasticity is depicted in Fig. 3, in which each curve shows the variation of weight as a function of the neuron's postsynaptic activity. The parameter that defines which curve is valid in each case is the value of the synaptic weight. According to Fig. 3, for higher values of the synaptic weight the curves are more elongated to the right.

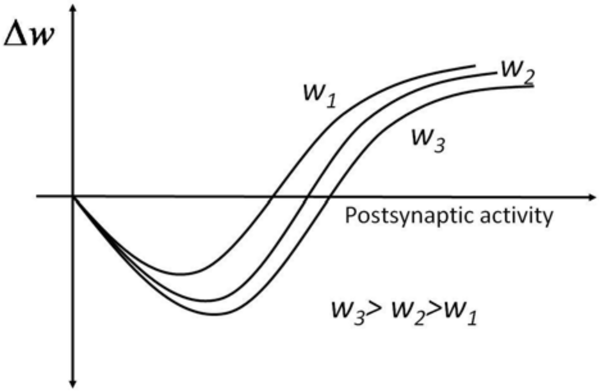


Fig. 3. Synaptic plasticity curves. The value of the initial weight determines which curve should be used for calculating weight variation. Curves with higher initial weights are more elongated to the right.

Metaplasticity regulates weight variation, down-regulating weight increment in synapses with high initial weights and up-regulating weight increment in synapses with low initial weights.

Intrinsic Plasticity. Although synaptic metaplasticity makes it difficult for synaptic weights to become either null or saturated, it does not totally preclude either of these two extreme situations. To eliminate the possibility of either weight annihilation or saturation, another important homeostatic property of real neurons should be taken into account: the so-called intrinsic plasticity.[1][9]

Intrinsic plasticity regulates the position (shift) of the neuron's activation function according to the past average level of activity in the neuron. The neuron's activation function is usually modeled as a sigmoidal function:

$$P(O) = \frac{1}{1 + e^{-25(\alpha - shift)}} \tag{3}$$

In which P(O) is the output probability of the neuron and a is the activation given by the sum of synaptic contributions.

Intrinsic plasticity was modeled according to the following equation, that yields the position of the sigmoid in terms of the previous position $shift_{t-1}$ and the previous sigmoid output O_{t-1}.

$$shift_t = \frac{\xi O_{t-1} + shift_{t-1}}{1 + \xi} \tag{4}$$

Parameter ξ is a small arbitrary factor ranging in our case from 0.1 to 0.0001. This equation means that the more the neuron keeps firing, the higher will be the rightward shift of the activation function, leading to a moderation of the neurons' firing probability in the future. Conversely, if the firing probability is low, the sigmoid will move leftwards, thereby increasing the probability of the neuron's firing in the future. Parameter ξ works as a learning factor. If we want quick convergence of the shifts with little interest in accuracy, we select a higher ξ. However if we are not worried about the duration of learning and we are mainly interested in accuracy, ξ must be set to a very small value.

Competition Between Neurons. Another neural mechanism used in our simulation, is competition between neurons. In biological neural networks, inhibitory neurons like basket or chandelier cells are able to reduce the activity of neighboring excitatory cell so that a few excitatory ones remain active. This competition allows to find the most activated neurons inside a neural pool. In our case the most activated neuron remains active while remaining ones are inactivated.

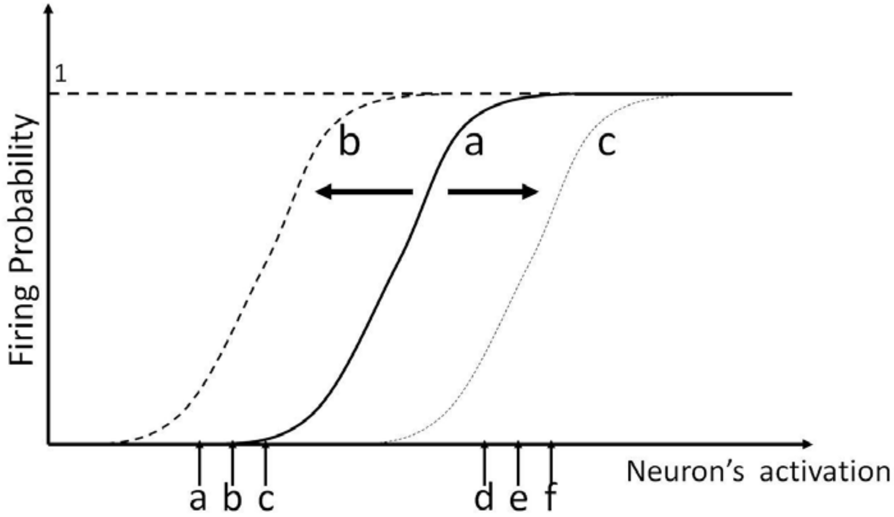


Fig. 4. According to intrinsic plasticity, low neuron activation (cases a,b,c) contributes to a leftward shift of the sigmoidal activation function. On the contrary high neuron activation (cases d,e and f) contributes to a rightward shift of this sigmoidal function.

3 Results with Different Kinds of Stimuli

As mentioned in the introduction, the purpose of our work is to apply the inclined floor stimulation technique to a robot for understanding how this technique contributes to accelerating crawling abilities in children.

Glenn Doman suggested placing the children on a ramp in a belly-down position. The inclination of the ramp should be increased up to the point where the movements of the child make his/her body slip on the ramp. For better results, parents are suggested to shout or clap when the child successfully moves down the ramp. Black and white stripes painted on the ramp contributes to improve even more the performance of the method, also helping to stimulate children sight.

Before placing the robot on the inclined floor its four artificial neurons fire in a random way, and the wheels (controlled by these neurons) move also randomly. Once the robot is placed on the ramp, it slides a little when it moves downward (sliding does not take place when the robot moves upward). Sliding makes the sensory input of the robot being highly activated and, since sliding happens during downward movements, downward movements produce a higher sensory input. In the case of the light sensor, the higher activation takes place because the sensor identifies more transitions between white and black stripes when sliding downward than when moving upwards. In the case of the ultrasonic detector it was modified to calculate accelerations, and accelerations are also more intense when the movement of the robot is helped by gravity, i.e when it slides downwards. The only stimulus in which gravity is not relevant is sound, which in our case is produced by a maraca. The maraca is played when the robot performs a correct movement.

Our working hypothesis is that, when a random sequence of neurons is such that makes the robot goes downwards, the subsequent sensory stimuli contribute to reinforce the weights between these neurons. In this way, different chains of neurons linked by strengthened weights are formed, each chain representing a specific way the robot goes downwards. The more intense the sensory stimuli, the stronger are the connections in a chain of neurons.

To evaluate these hypotheses we counted the number of iterations (each iteration corresponding to each movement of the robot) until the robot learns to go downward under several arrangements of its environment. For example, a specific arrangement of the environment is the robot placed on the ramp (so that it experiments body acceleration) without sound and without stripes painted on the ramp (the behavior of the robot with some of these arrangements can be seen on youtube by placing the authors' family names in the youtube searching tool). We consider that the robot has learned when it moves in the same direction during five consecutive iterations.

In Table 1 we show different arrangements of the robot environment in which different types of sensory stimuli were used. We performed 15 tests for each configuration. In each test we counted the number of iterations that it takes for the robot to learn. The mean and standard deviation of the 15 tests of each configuration are also shown in Table 1.

4 Practical Implications

According to Table 1, when two stimuli are combined, the number of iterations it takes for the robot to learn is less than when a single stimulus appears alone.

Table 1. Number of iterations that takes for the robot to learn. Each iteration corresponds to one movement of the robot.

	Mean	Standard Deviation
Stripes	32.00	11.15
Sound	21.69	5.63
Acceleration	27.53	4.15
Stripes+Sound	19.23	3.57
Sound+Acceleration	21.15	3.44
Stripes+Acceleration	24.07	6.72
Stripes+Sound+Acceleration	19.38	2.99

In the case of having three stimuli, the number of iterations that it takes for the robot to learn is less than when having two or one stimuli. This result is consistent with Doman's methods in which parents are suggested to combine all these stimuli for improving the methods' efficacy.

It can be also noticed that the presence of sound (the noise of a maraca when the robot performs correct movements) yields the lower number of iterations, either when the sound is alone or combined with different kinds of stimuli. This happens because the intensity of sound is always the same, regardless of how much the robot slides on the floor (which depends on the circumstances like the adjustment of the robot brakes and the slipperiness and inclination of the ramp). According to our studies, sound seems to be an important stimulus for stimulating crawling in the robot and, hypothetically in children. Clapping, shouting or even playing a musical instrument would help children to consolidate the successful motor commands that make them crawl downwards on the inclined floor. All these suggestions should be empirically tested with children.

5 Conclusion

In this work we have constructed a very simple robotic structure with three type of sensors and two independent wheel sets. A fully interconnected four-neurons neural network receive sensory inputs and controls the wheel sets. Each neuron performs a single movement of a specific wheel set. For example, when neuron 1 fires, it only produces clockwise movements in the frontal wheel set. This robotic structure was placed in an inclined floor to mimic Doman's method for stimulating crawling abilities in children. Other stimuli like sound or alternating black and white stripes painted on the ramp are also added to the robotic environment for the purpose of mimicking the arrangement of the environment in Doman's method.

Cybernetic principles envisioned by Doman's team are present in the way our robotic neural network works. For example, the presynaptic rule (used for synaptic weight adjustment) and our rule for modeling intrinsic plasticity are consistent with cybernetic and biological principles.

We constructed the robot using a LEGO Mindstorms NXT toolkit and programmed the neural network using the RWTH Mindstorms NXT toolbox of Matlab (which is freely available on the internet).

Despite the simplicity of the structure and the small number of neurons, the robot is capable of continuous learning, that is, it learns during the execution of its movements. In this way the robot learned what sequences of motor commands lead him downwards and what sequences failed to do this.

For simulating the different stimuli that are present in Doman's inclined floor technique, different stimuli were applied to the robotic structure. We performed 15 tests with each stimulus and calculate the average number of iterations it takes for the robot to learn. According to our expectations and consistently to Doman's empirical results, learning was quicker when more rewarding stimuli were present. Our work also suggests some improvements for the inclined floor technique that should be empirically tested.

References

1. Desai, N.S., Rutherford, L.C., Turrigiano, G.G.: Plasticity in the Intrinsic Excitability of Cortical Pyramidal Neurons. Nature Neurosciences 2(6), 515–520 (1999)
2. Desai, N.: Homeostatic Plasticity in the CNS: Synaptic and Intrinsic Forms. Journal of Physiology 97, 391–402 (2003)
3. Doman, G.: What to do about your brain-injured child. In: Or your brain-damaged, mentally retarded, mentally deficient, cerebral-palsied, emotionally disturbed, spastic, flaccid, rigid, epileptic, autistic, athetoid, hyperactive, downs child. The Better Baby Press, Philadelphia (1990)
4. Doman, G., Doman, D., Hagy, B.: How To Teach Your Baby To Be Physically Superb: From Birth To Age Six; The Gentle Revolution. Square One Publishers (2006)
5. Haykin, S.: Neural Networks: a comprehensive foundation. Prentice Hall, Inc., Englewood Cliffs (1999)
6. Hebb, D.O.: The organization of behaviour. Wiley, New York (1949)
7. Manabe, T., Shinoe, T., Matsui, M., Taketo, M.: Modulation of Synaptic Plasticity by Physiological Activation of M1 Muscarinic Acetylcholine Receptors in the Mouse Hippocampus. The Journal of Neuroscience 25(48), 11194–11200 (2005)
8. Peláez, F.J.R., Simões, M.G.: A computational model of synaptic metaplasticity. In: Proceedings of the IJCNN 1999 International Joint Conference of Artificial Neural Networks, Washington D.C (1999)
9. Peláez, J.R., Piqueira, J.R.C.: Biological clues for up-to-date artificial neurons. In: Andina, D., Phan, D.T. (eds.) Computational Inteligence: for Engineering and Manufacturing, 1st edn., vol. 1, pp. 1–19. Springer-Verlag, Berlin (2006)
10. Takeda, N., Horii, A., Mochizuki, T., Okakura-Mochizuki, K., Yamatodani, A.: Effects of vestibular stimulation on acetylcholine release from rat hippocampus: an in vivo microdialysis study. Journal of Neurophysiology 72, 605–611 (1994)
11. Wiener, N.: Cybernetics: or control and communication in the animal and the machine. MIT Press, Cambridge (1948)

On the Biological Plausibility of Artificial Metaplasticity*

Diego Andina and Javier Ropero-Peláez

Group for Automation in Signal and Communications, Technical University of Madrid, Spain and
Federal University of A.B.C, Brazil

Abstract. The training algorithm studied in this paper is inspired by the biological metaplasticity property of neurons. Tested on different multidisciplinary applications, it achieves a more efficient training and improves Artificial Neural Network Performance. The algorithm has been recently proposed for Artificial Neural Networks in general, although for the purpose of discussing its biological plausibility, a Multilayer Perceptron has been used. During the training phase, the artificial metaplasticity multilayer perceptron could be considered a new probabilistic version of the presynaptic rule, as during the training phase the algorithm assigns higher values for updating the weights in the less probable activations than in the ones with higher probability.

1 Introduction

Artificial Metaplasticity (AMP) term was first introduced by Andina *et al* [1] for an Artificial Neural Network (ANN) of the Multilayer Perceptron type (MLP), referred as AMMLP. During the AMMLP training phase, the matrix weight W that models the synaptic strength of its artificial neurons is updated according to the probability of the input patterns and therefore of the corresponding synaptic activations. The concept of biological metaplasticity was defined in 1996 by W.C. Abraham [2] and now is widely applied in the fields of biology, neuroscience, physiology, neurology and others [2,3,4]. The prefix "meta" comes from Greek and means "beyond" or " above". In neuroscience and other fields "metaplasticity" indicates a higher level of plasticity, expressed as a change or transformation in the way synaptic efficacy is modified. Metaplasticity is defined as *the induction of synaptic changes, that depends on prior synaptic activity [3,5].* Metaplasticity is due, at least in part, to variations in the level of postsynaptic depolarization that induce synaptic changes. These variations facilitate synaptic potentiation and inhibit synaptic depression in depressed synapses (and vice versa in potentiated synapses). The direction and the degree of the synaptic alteration are functions of postsynaptic depolarization during synaptic activation. Upregulation -incrementing, reinforcement of synaptic efficacy- is termed

[*] This research has been supported by Group for Automation in Signal and Communications, GASC/UPM.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 119–128, 2011.

long-term potentiation (LTP), whereas downregulation -decrementing inhibing- is known as long-term depression (LTD). LTP and LTD are believed to be fundamental to storage of memory in the brain and hence learning.

The induction of synaptic changes in the levels of neural activity is explained [6] in Fig. 1. Metaplasticity can be represented as variations in curve elongation with respect to the level of activity and implies a shift of the LTP threshold according to the weight strength of the synapse [6]. Fig. 1 graphically illustrate this idea. Understanding metaplasticity may yield new insights into how the modification of synapses is regulated and how information is stored by synapses in the brain. [7,8,9]

This paper is organized as follows. Section 2 provides a brief introduction to related concepts (e.g. synaptic plasticity). Main postulation regarding the relation between metaplasticity and Shannon's Information Theory is introduced in Section 3, to ease the understanding of the proposed model. In Section 4, general mathematical theory is applied to describe the proposed implementation of AMP in a MLP, whose learning process is based on error minimization. In Section 5, we implement the AMP algorithm in the MLP neural network, which is trained with the BP algorithm with a modified objective function. Section 6 presents a discussion on the biological plausibility of the AMMLP algorithm, refering to experimental results. Finally, Section 7 summarizes main conclusions.

2 Synaptic Plasticity and Metaplasticity

Synaptic plasticity refers to the efficacy modulation of information transmission between neurons, being related to the regulation of the number of ionic channels in synapses. Synaptic plasticity mechanisms involve both molecular and structural modifications that affect synaptic functioning, either enhancing or depressing neuronal transmission. They include redistribution of postsynaptic receptors, activation of intracellular signaling cascades, and formation/retraction of the dendrites [10]. The first model of synaptic plasticity was postulated by Hebb and it is known as the Hebb rule [11].

In Fig.1, the effect of metaplasticity is illustrated. This graphic shows a family of curves in which each curve indicates the biological variation in weight, $\Delta\omega$, respective of the neurons activation frequency or postsynaptic activity. If postsynaptic activity is high, by metaplasticity property, the curve will move to the right, reinforcing the LTP. Andina postulates [1] that high postsynaptic activity corresponds to high frequent excitations -frequent input classes in an artificial model-. In the same way, the left-hand side curves of the family corresponding to low previous synaptic activity correspond to low frequent excitations produced by patterns of unfrequent classes. During training, postsynaptic activity is the same for each training pattern -one excitation by epoch-. As it can be observed in Fig.1, for a given postsynaptic activity value, a higher $\Delta\omega$ corresponds to the un-frequent classes curves than to the curves corresponding to frequent ones.

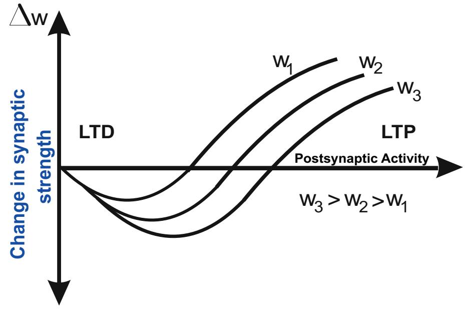


Fig. 1. Heterosynaptic BCM model [21] illustrates changes in synaptic strength due to postsynaptic activity in biological neurons

.

2.1 Intrinsic Plasticity

A third concept about biological plasticity is the intrinsic plasticity, and it must not be misunderstood with the previous ones. When neurons are receiving and responding to synaptic inputs, the synaptic metaplasticity makes it difficult for synaptic weights to become either null or saturated. But the metaplasticity property cannot fully avoid these two extreme situations. For totally precluding the possibility of either weight annihilation or saturation, another important homeostatic property of real neurons should be taken into account: the so-called intrinsic plasticity [12]. Intrinsic plasticity regulates the position (rightward shift) of neurons activation function, according to previous levels of activity [6]. Intrinsic plasticity is not modelled in the AMMLP.

3 Metaplasticity and Shannon Information Theory

As is well-known within the ANN field, in 1949 Hebb postulated that during the learning phase, synaptic connections between biological neurons are strengthened due to the correlated activity of presynaptic and postsynaptic neurons [11]. This plasticity property of synaptic connections is modeled in many ANNs as a change in the connection weights of the artificial neurons or nodes. Therefore, synaptic plasticity of biological neural networks has been simulated in artificial networks by changing the weight values of the simulated neuronal connections. These weights are the most relevant parameters in ANN learning and performance. Modeling these new discovered properties of biological neurons that follow metaplasticity rules provides a large potential for improving ANN learning. In addition, the results of these simulations may also corroborate the biological hypothesis of neuronal metaplasticity. Utilizing the potential of this new modeling approach, artificial metaplasticity (AMP) models have been devised and tested. A model that closely followed biological metaplasticity and intrinsic plasticity was successfully tested in the reinforcing the calcium dysregulation hypothesis for Alzheimer's disease [13]. However, of all AMP models tested by the authors, the most efficient model (as a function of learning time and performance) is the approach that connects metaplasticity and Shannon information theory, which establishes that less frequent patterns carry more information than frequent patterns [14]. This model then defines artificial metaplasticity as a learning procedure that produces greater modifications in the synaptic weights with less frequent patterns than frequent patterns, as a way of extracting more information from the former than from the latter. As Biological metaplasticity, AMP then favors synaptic strengthening for low-level synaptic activity, while the opposite occurs for high level activity. The model is applicable to general ANNs. Andina *et al.* propose general AMP concepts for ANNs, and demonstrate them over Radar detection data [1].

4 Backpropagation Algorithm and AMP

The AMP implementation applied tries to improve results in learning convergence and performance by capturing information associated with significant rare events. *It is based on the idea of modifying the ANN learning procedure such that un-frequent patterns which can contribute heavily to the performance, are considered with greater relevance during learning without changing the convergence of the error minimization algorithm.* It is has been proposed on the hypothesis that biological metaplasticity property maybe significantly due to an adaptation of nature to extract more information from un-frequent patterns (low synaptic activity) that, according to Shannon's Theorem, implicitly carry more information.

4.1 Mathematical Definitions

Let us define an input vector for a MLP with n inputs (bias inputs are assumed to exist and be of fixed value set to 1): $x \in R^n$, where R^n is the n-dimensional space, i.e. $x = (x_1, x_2, ..., x_n)$, $x_i \in R^1$, $i = 1, 2, ..., n$; and its corresponding j outputs given by vector $y = (y_1, y_2, ..., y_n)$, $y_i \in (0, 1)$, $j = 1, 2, ..., m$ [15]. Let us consider now the random variable of input vectors $X = (X_1, X_2, ..., X_n)$ with probability density function (pdf) $f_X(x) = f_X(x_1, x_1, ..., x_n)$. The strategy of MLP learning is to minimize an expected error, E_M, defined by the following expression:

$$E_M = \varepsilon \{E(x)\} \tag{1}$$

where $E(x)$ is the expression of an error function between the real and the desired network output, being respectively $Y = F(X)$, with pdf $f_Y(y)$ and Y_d the desired output vector, and $F(X)$ is the nonlinear function performed by the MLP. The symbol ε represents the mathematical expectation value, that is,

$$E_M = \int_{R^n} E(x) f_X(x)\, dx \tag{2}$$

4.2 AMP in Gradient Descent Algorithm

BackPropagation training algorithm applied in MLPs follows Widrow gradient descent algorithm over an estimation of this expected error in each training iteration, $t \in N$, for determining the necessary modification in the ANN weight matrix $W(t)$ in each bias and weight value in the MLP [15]. The algorithm objective is to reduce the output classification error in subsequent training epochs, stopping the training phase if the error is low enough to satisfy the design requirements.

To introduce AMP in the gradient descent algorithm, let us define

$$e(x) \equiv E(x) f_X(x)$$

and manipulate Eq. (2) in the following way:

$$\begin{aligned} E_M &= \int_{R^n} \frac{e(x) f_X^*(x)}{f_X^*(x)} dx = \\ &= \varepsilon^* \left\{ \frac{e(x)}{f_X^*(x)} \right\} \end{aligned} \qquad (3)$$

where a new probability density function (pdf) $f_X^*(x)$ has been introduced, requiring that $f_X^*(x) \neq 0$ wherever $e(x) \neq 0, \forall x \in R^n$ and new mathematical expectation, ε^*, defined in Eq. (3) represents that the minimization of E_M can also be achieve from statistical inference theory applied to Eq. (3), by estimating over the weighted function $e(x)/f_X^*(x)$ instead of $e(x)$, under $f_X^*(x)$ pdf, through the estimator $\hat{E}_M$, defined as:

$$\hat{E}_M \equiv \frac{1}{P} \sum_{k=1}^{P} \frac{e(x_k^*)}{f_X^*(x_k^*)} \qquad (4)$$

where $x_k^*, k = 1, 2, ..., P$, are independent sample vectors whose pdf is the weighting function $f_X^*(x)$. Note that many functions may fix to the definition of $f_X^*(x)$, in particular:

$$[f_X^*(x)]_{opt} = \frac{1}{E_M} e(x) \qquad (5)$$

that can be proved by taking Eq. (5) into Eq. (4); only one simple sample vector $(P = 1)$ is then required for exactly estimating E_M without error. The optimal solution for $f_X^*(x)$ given by Eq. (5) is not realizable, because E_M is not known *a priori* (it has to be estimated by Eq. (4)). But, a suboptimal solution can be used. For example, a common suboptimal solution for $f_X^*(x)$ applied and tested succesfully in many applications is:

$$f_X^*(x) = \frac{A}{\sqrt{(2\pi)^N}.e^{B \sum\limits_{i=1}^{N} x_i^2}} \equiv \frac{1}{w_X^*(x)} \qquad (6)$$

where $w_X^*(x)$ is defined as $1/f_X^*(x)$, N is the number of neurons in the MLP input layer, and parameters A and $B \in R^+$ are algorithm optimization values which depend on the specific application of the AMLP algorithm. Values for A and B have been empirically determined. Eq. (6) is a gaussian distribution, so it has been assumed that X pdf is Gaussian (if it is not the case, the real X pdf should be used instead). Then, $w_X^*(x)$ has high values for un-frequent x values and close to 1 for the frequent ones and can therefore be straightforwardly applied in weights updating procedure to model the biological metaplasticity during learning.

5 AMP in MLP Training: AMMLP

In the case of an MLP trained with BPA applied to L classes, H_l, $l = 0, 1, ..., L-1$, previous studies have shown that the output for each class is the MLP inherent estimation of *a posteriori* probability of the class [16], based on Bayes Theorem, we then have:

$$y_l \cong P\left(H_l \left| x \right.\right) = \frac{f_X\left(x \left| H_l \right.\right) .P\left(H_l\right)}{f_X\left(x\right)} \tag{7}$$

This enables a direct implementation of metaplasticity. For each class, by assuming the proposed AMP model described in subsection 4.2 can be make $f_X^*(x) = f_X(x)$ and from Eq.(7) and Eq. (4)

$$e\left(x \left| H_l \right.\right) E\left(x\right) f_X\left(x \left| H_l \right.\right)$$

$$\begin{aligned} \hat{E}_{M_l} &= \frac{1}{M_l} \sum_{k=1}^{M_l} \frac{E\left(x_k\right) f_X\left(x \left| H_l \right.\right)}{f_X\left(x_k\right)} \\ &\cong \frac{1}{M_l} \sum_{k=1}^{M_l} E\left(x_k\right) \frac{y_l}{P\left(H_l\right)} \end{aligned} \tag{8}$$

where $k = 1, 2..., M_l$, are the independent sample vectors of class l in the training set. Then, from Eq. (8) and Eq. (4)

$$\frac{y_l}{P\left(H_l\right)} \cong \frac{1}{f^*\left(x\right)} \tag{9}$$

Eq. (7) takes advantage of the inherent *a posteriori* probability estimation for each input class of MLP outputs, so it is used to quantify a pattern's frequency. Note that if this is not the case, as it happens in first steps of BPA training algorithm, the training may not converge. In this first steps, the outputs of the MLP does not provide yet any valid estimation of the *a posteriori* probabilities, but rather random values corresponding to initial guess of the MLP weights, W . It is then better in these first steps of training, either to apply ordinary BPA training or to use another valid weighting function till BPA starts to minimize the error objective. Also, many suboptimal functions may yield good results. For example, in the following experiments, a typical approximation premise that assumes a Gaussian distribution for the inputs has been implemented, proposing the function for weight updating (known as a *weighting function*) [1], given by Eq. (6).

To analytically introduce AMP in an arbitrary MLP training, all that has to be done is to introduce the weighting function in the error function between the real and the desired network output, as a function of the weights matrix $W(t)$ in each training iteration, t, that is

$$E^{*}\left[W(t)\right]=\frac{E\ \left[W(t)\right]}{f_{X}^{*}(x)} \tag{10}$$

And apply the BPA [15] to the weighted error $E^{*}(W)$ for weights reinforcement in each iteration $t \in$ N. If s, j, $i \in N$ are the MLP layer, node and input counters respectively, for each $W(t)$ component, $w_{ij}^{(s)}(t) \in R$, and being $\eta \in R^{+}$ the *learning rate*, then the weight reinforcement in each iteration is given by:

$$\begin{aligned} w_{ij}^{(s)}(t+1) &= w_{ij}^{(s)}(t)-\eta\frac{\partial E^{*}\left[W(t)\right]}{\partial w_{ij}^{(s)}} \\ &= w_{ij}^{(s)}(t)-\eta\frac{1}{f_{X}^{*}}\frac{\partial E\left[W(t)\right]}{\partial w_{ij}^{(s)}} \end{aligned} \tag{11}$$

So, as the pdf weighting function proposed is the distribution of the input patterns that does not depend on the network parameters, the AMMLP algorithm can then be summarized as a weighting operation for updating each weight in each MLP learning iteration as:

$$\varDelta^{*}w=w^{*}(x)\,\varDelta w \tag{12}$$

being $\varDelta w=w(t+l)-w(t)$ the weight updating value obtained by usual BPA and $w^{*}(x)$ the realization of the described weighting function $w^{*}(x)$ for each input training pattern x. During the training phase, the artificial metaplasticity multilayer perceptron could be considered a new probabilistic version of the presynaptic rule, as during the training phase the algorithm assigns higher values for updating the weights in the less probable activations than in the ones with higher probability.

6 Discussion

Plasticity and Metaplasticy still pose a considerable challenge for research in terms of experimental design and interpretation [22]. Along the years, different mathematical models of synaptic computation have been proposed. In the classical Hebb model [11], the curve relating the increment of synaptic weight to postsynaptic activation is a straight line without synaptic depression. In Sejnowski's covariance model [23, 24], regions of potentiation and depression are separated by a LTP threshold. In [25] Abraham and Bear consider it as a homosynaptic property (i.e. involving only the synapse under study, without the need of considering the influence of nearby synapses). Whereas the model by Bienenstock, Cooper and Munro [21] yields a curve that is closer to reality, with LTP threshold determined by postsynaptic activation and without LTD threshold. In the BCM model, the LTP threshold is the same for all neuron synapses, so that metaplasticity would affect even non-active synapses (heterosynaptic plasticity). But despite its unquestionable biological characteristics, the BCM model, cannot be regarded to be the ultimate model of synaptic metaplasticity. According to Mockett and colleagues [27] metaplasticity is inherently a homosynaptic phenomenon in contrast to the heterosynaptic nature of the BCM rule. Finally, Artola, Bröcher and Singer's [28]) extended model (ABS model) is not analytical, as those just discussed, but is based on empirical experimental data. In the ABS model, LTP and LTD thresholds shift to lower values for higher levels of the activation of neighbouring synapses.

In its implementation characteristics, the proposed AMP model follows closer to the BCM model. We do not pretend to determine its superiority, but neurobiology inspires computer science and vice versa, and we report that the empirical results of AMP show a great potential, in terms of *improving learning* and therefore performance in most cases, no matter in what multidisciplinary application is applied [1,17,18,28].

7 Conclusion

We describe and discuss the biological plausibility of an artificial model of metaplasticity, a relevant property of neurons. Tested on different multidisciplinary applications, it achieves a more efficient training and improves Artificial Neural Network Performance. The model follows the BCM heterosynaptic biological model. During the training phase, the artificial Artificial Metaplasticity Multilayer Perceptron could be considered a new probabilistic version of the presynaptic rule, as it assigns higher values for updating the weights in the less probable activations than in the ones with higher probability.

References

1. Andina, D., Alvarez-Vellisco, A., Jevtic, A., Fombellida, J.: Artificial metaplasticity can improve artificial neural network learning. In: Intelligent Automation and Soft Computing, SI on Signal Processing and Soft Computing, vol. 15, pp. 683–696 (2009), ISSN: 1079-8587
2. Abraham, W.C.: Activity-dependent regulation of synaptic plasticity (metaplasticity) in the hippocampus. In: Kato, N. (ed.) The Hippocampus: Functions and Clinical Relevance, pp. 15–26. Elsevier, Amsterdam (1996)
3. Abraham, W.C., Bear, M.F.: Metaplasticity: The plasticity of synaptic plasticity. Trends in Neurosciences 19, 126–130 (1996), doi:10.1016/S0166-2236(96)80018-X

4. Kinto, E., Del-Moral-Hernandez, E., Marcano-Cedeño, A., Ropero-Pelaez, J.: A preliminary neural model for movement direction recognition based on biologically plausible plasticity rules. In: Proc. 2nd Int. Work-Conf. on the Interplay between Natural and Artificial Computation, Lecture Notes in Computer Science, pp. 628–636. Springer, Berlin (2007)
5. Chiappalone, M., Vato, A., Berdondini, L., Koudelka, M., Martinoia, S.: Network Dynamics and Synchronous Activity in Cultured Cortical Neurons. Int J. Neural Syst. 17(2), 87–103 (2007)
6. Ropero-Pelaez, J., Piqueira, J.R.: Biological clues for up-to-date artificial neurons. In: Computational Intelligence for Engineering and Manufacturing, pp. 131–146. Springer, Heidelberg (2007)
7. Bolle, D., Heylen, R.: Adaptive Thresholds for Neural Networks with Synaptic Noise. Int J. Neural Syst. 17(4), 241–252 (2007)
8. Abraham, W.C.: Metaplasticity: tuning synapses and networks for plasticity. Nature Reviews Neuroscience 9, 387–399 (2008), doi:10.1038/nrn2356.
9. Daoudal, G., Debanne, D.: Long-Term Plasticity of Intrinsic Excitability: Learning Rules and Mechanisms. Learning & Memory. Nature Reviews Neuroscience 10, 456–465 (2003), doi:10.1101/lm.65303
10. Neves, G., Cooke, S.F., Bliss, T.V.: Synaptic plasticity, memory and the hippocampus: a neural network approach to causality. Nature Rev. Neurosci. 9, 65–75 (2008), doi:10.1038/nrn2303.
11. Hebb, D.O.: The Organization of Behavior. In: Mahwah, N.J. (ed.) Reedition of the 1949 original (2002) ISBN-10: 0805843000, ISBN-13: 978-0805843002
12. Cudmore, R.H., Turrigiano, G.G.: Long-Term Potentiation of Intrinsic Excitability in LV Visual Cortical Neurons. Journal Neurophysiology 92, 341–348 (2004), doi:10.1152/jn.01059.2003.
13. Monteiro, J.L., Lobo-Netto, M., Andina, D., Pelaez, J.R.: Using Neural Networks to Simulate the Alzheimer's Disease. In: Davis, C.G., Yeh, R.T. (eds.) Proc. World Automation Congress, Hawaii, HI, USA, pp. 1–6 (2008) ISBN: 978-1-889335-38-4. INSPEC Accession Number: 10411864
14. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal 27, 379–423 (1948)
15. Andina, D., Pham, D.T. (eds.): Computational Intelligence for Engineering and Manufacturing. Springer, Heidelberg (2007)
16. Rucky, D.W., Rogers, S.K., Kabrisk, M., Oxley, M.E., Suter, B.W.: The multi-layer perceptron as an approximation to a Bayes optimal discrimination function. IEEE Transactions on Neural Networks 1, 296–298 (1990), doi:10.1109/72.80266.
17. Marcano-Cedeño, A., Álvarez-Vellisco, A., Andina, D.: Artificial metaplasticity MLP applied to image classification. In: Proc. 7th Int. Conf. on Industrial Informatics, Cardiff, United Kingdom, pp. 650–653 (2009), doi:10.1109/INDIN.2009.5195879.
18. Marcano-Cedeño, A., Quintanilla-DomÍnguez, J., Andina, D.: Wood Defects Classification Using Artificial Metaplasticity Neural Network. In: Proc. 35th Annua Conf. on of the IEEE Industrial Electronics Society, Porto, Portugal, pp. 3422–3427 (2009), doi:10.1109/IECON.2009.5415189
19. De Long, E., Clarke-Pearson, D.: Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. Biometrics 44, 837–845 (1988)
20. Egan, J.: Signal Detection Theory and ROC analysis. In: Series in Cognition and Perception, SAcademic Press, New York (1975)

21. Bienestock, E.L., Cooper, L.N., Munro, P.W.: Theory of the developement of neuron selectivity: orientation specifity and binocular interaction in visual cortex. J. Neurosci. 2(1), 32–48 (1982)
22. Abraham, W.C.: Metaplasticity: tuning synapses and networks for plasticity. In: Nature Reviews/neuroscience, vol. 9, pp. 387–399. Nature Publishing Group (May 2008)
23. Sejnowski, T.J.: Storing covariance with non linearly interacting neurons. J. Math. Biol. 4, 303–321 (1977)
24. Sejnowski, T.J., Chattarji, S., Stanton, P.K.: Homosynaptic long-term depression in hippocampus and neocortex. Seminars in the Neurosciences 2, 355–363 (1990)
25. Abraham, W.C., Bear, W.C.: Metaplasticity the plasticity of synaptic plasticity. Trends Neurosci. 19, 126–130 (1996)
26. Mockett, B., Coussens, C., Abraham, W.C.: NMDA receptor-mediated metaplasticity during the induction of long-term depression by low-frequency stimulation. European Journal of Neuroscience 15, 1819–1826 (2002)
27. Artola, A., Singer, W.: Long-term depression of excitatory synaptic transmission and its relationship to long-term potentiation. Trends Neurosci. 16, 480–487 (1993)
28. Marcano-Cedeño, A., Quintanilla-Domínguez, J., Andina, D.: Breast Cancer Classification applying Artificial Metaplasticity Algorithm. In: Neurocomputing, Corrected Proof, pp. 925–2312. Elsevier, Amsterdam (2010), (in press) doi:10.1016/j.neucom.2010.07.019

Dynamic Model of the dLGN Push-Pull Circuitry

Rubén Ferreiroa[1], Eduardo Sánchez[1], and Luis Martínez[2]

[1] Grupo de Sistemas Intelixentes (GSI)
Departamento de Electrónica e Computación,
Universidade de Santiago de Compostela,
15782 Santiago de Compostela, Spain
rferreiroa@gmail.com, eduardo.sanchez.vila@usc.es
[2] Visual Neuroscience
Instituto de Neurociencias de Alicante,
03550 Alicante, Spain
l.martinez@umh.es
http://www-gsi.usc.es/index.html

Abstract. In the present work we propose a dynamic model of the lateral geniculate nucleus (dLGN) that allows the implementation of different configurations of the push-pull circuitry in order to study the spatio-temporal filtering being carried out. It is widely accepted that each relay neuron receives only one input from a single retinal ganglion cell, which leads to interpret that the thalamus preserves the retinal structure of the receptive field and thus works as a simple relay station. We believe that this assumption is not fully valid and the thalamus could perform a more relevant processing of information through its complex push-pull circuitry. To test this hypothesis, a computational model was developed with a wiring configuration (convergence/divergence) between the retina and the dLGN based on experimental evidences, and a realistic description of the ON and OFF channels of dLGN. We found that this configuration may help improve the contrast of a stimulus by increasing its synaptic weight on higher frequencies.

1 Introduction

Visual information is processed through multiple pathways that originate in different types of retinal ganglion cells and remain relatively well segregated through the dorsal lateral geniculate nucleus (dLGN) until reaching the visual cortex (Cleland and Lee 1985; Cleland et al. 1971a; Hamos et al. 1987; Mastronarde 1992; Sincich et al. 2007; Usrey et al.1999). In the cat's visual system, there are two main pathways (X and Y) that differ in their response properties and axonal projections. Whereas most X retinal afferents project to a single dLGN layer (layer A), Y retinal afferents can diverge into two dLGN layers (layer A, C). In our study we will focus on X-type afferents located in the fovea, responsible for processing the details of the scene, and the DLGN's layer A, which is composed by two types of neurons: relay cells and interneurons. In addition

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 129–137, 2011.

to the retinal inputs, geniculate relay cells are innervated by non-retinal inputs that account for approximately 90%-95% of the total synaptic inputs. They come from local GABAergic neurons, reticular cells, interneurons, and also from two sources of extrinsic inputs, i.e a feedback projection from cortical layer 6 and an ascending projection from various scattered cell groups in the brainstem reticular formation. This complex circuitry seems to play a role that could be more relevant that just being a relay step. Besides the complicate schema of the inputs, we should take into account that the number of relay cells in the dLGN of the cat doubles the number of retinal ganglion cells [1]. Moreover, inhibition in the dLGN does not come from the retina, but locally from the interneurons that receive inputs from retinal ganglion cells of the opposite sign, and it was reported that the number of interneurons is half of the number of retinal ganglion cells [1]. The issue about how many connections are established by each relay cell with the retina and how many with the local interneurons, was addressed by different authors (Pei et al.,1991; Hirsch et al.,1995; Martinez et al.,2005), who carried out whole cell recordings in the dLGN of the cat while stimulating with sparse-noise protocol (Jones and Palmer,1987). With this kind of recordings, ON and OFF maps(receptive field maps when stimulated with a white and dark square) were obtained. By comparing both maps, two main mismatches were detected. First, the center of the ON map is bigger than the center of the OFF map, this difference being measured with the size index(SI). Second, their peaks are displaced, this feature being measured with the overlap index (OI). The differences between the ON and the OFF map represent a good opportunity to know more about the connectivity of the system as the two maps come from different pathways. While the ON map come directly from the retinal ganglion cells, the inhibition comes through the interneurons. In order to know which are the connectivity configurations (convergence/divergence) that best fit the data obtained in whole cell recordings, Molano and Martinez (2009) [14] built a retino-thalamic computational model similar to [2] from which we can obtain the number of connections between system elements that best fit the experimental data. The values obtained are consistent with those deduced by other authors using a different methodology [3].

2 The Problem

The goal of this paper is to explore the type of spatio-temporal processing that is taking place in the dLGN, with the convergence/divergence data of the push-pull circuit derived from the computational model developed by Molano and Martinez (2010) [14]. Concretely, we face the following questions: 1. Center-surround receptive fields from channel ON are sufficiently effective in edge detection? 2. What makes the asymmetry of the ON and OFF channels and the divergence/convergence of information processing that the retina sends to the dLGN? 3. Could the push-pull circuitry increase the spatial resolution?.

3 Computational Methods

3.1 Model Composition

The retina is represented by two rectangular layers of 40x40 pixels, the first one representing the ON type ganglion cells, the second one the OFF type ganglion cells. The lateral geniculate nucleus is modeled by a rectangular layer with dimensions 40x40 pixels, each point in the lattice having two neuron models, one for the relay neurons and the other for interneurons. The fraction of neurons per point is 1:4, meaning that for each interneuron there are four neighbor relay cells (Molano and Martinez, 2009) [14] (see Figure 1).

3.2 Coordinate System and Topology

The retina and the thalamus are simulated by means of a lattice of size 40x40 pixels that represents a patch with a parafoveal visual field of 8°x8°. Each position in the layer corresponds to a node, each node can define subnets of neurons. In our case, each node of the layer of the thalamus have two subnets, one formed by one interneuron and one consisting of four relay neurons.

3.3 Connectivity

Each relay cell in the thalamus is first connected to its nearest neighbor in the retinal lattice, i.e. the one from which its polarity is inherited. The probability of each thalamic cell, re-centered at retinal coordinates at the position of its first retinal input (X), being connected to another retinal ganglion cell is modeled as a Gaussian function of their relative distance [4]. The synaptic strength of the connections is also assumed to follow a Gaussian distribution of the distance between the receptive-field centers [4]. The function for both connection probability and strength is as follows (see Figure 1):

$$P = p_{max} e^{-(x-y)^2/2\sigma_{con}^2} \tag{1}$$

3.4 Input and Output Models

The stimuli used in our simulations are simple geometric figures as bright circles, rectangles and bars. Stationary stimuli are mathematically represented by using bidimensional arrays of size 40x40, where the coefficients represent either high or low light intensity.

The spatial arrangement of the receptive fields of retinal ganglion cells are captured by a difference-of-Gaussians model (Rodieck, Enroth-Cugell and Robson, 1966) in which the spatial receptive field is expressed as:

$$D(x,y) = (\frac{1}{2\pi\sigma_{cen}^2} e^{-(x-y)^2/2\sigma_{con}^2} - \frac{B}{2\pi\sigma_{sur}^2} e^{-(x-y)^2/2\sigma_{sur}^2}) \tag{2}$$

The first Gaussian function describes the center and the second one the surround receptive field, the sizes of both fields being determined by the parameters σ_{cen}

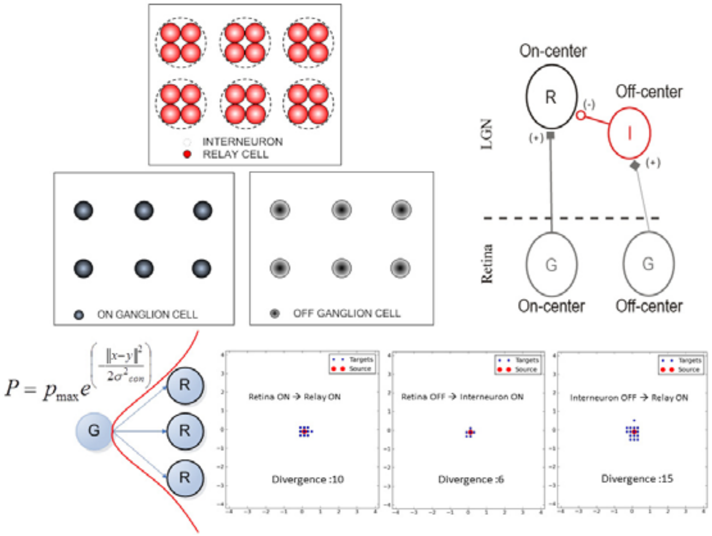


Fig. 1. Topology and spatial connectivity of the LGN model. Arrangement of neurons in the lattice. Circuit Design of dLGN and number of divergence connections between different layers.

and σ_{sur}. Receptive field parameters for the ganglion cells have been set in accordance to Allen and Freeman (2006). The rate of the pulse train evoked by a stimulus in the ganglion cells is modeled with a 2D convolution function:

$$r = r_0 + \int\int D(x,y)s(x,y)dxdy \tag{3}$$

where r represents the firing rate evoked by a stimulus $s(x,y)$, r_0 is the background firing, and $D(x,y)$ the receptive field of ganglion cells.

Finally, the firing rate of each ON/OFF ganglion cell modulates a pulse train generator following the statistics of a homogeneous Poisson process:

$$P_T(n) = \frac{rT}{n!}e^{-rT} \tag{4}$$

Figure 2 provides an overview of all the components involved in our model.

3.5 Neuron Model

We use an implementation of the Adaptative Exponential Integrate and Fire (AEIaF) neuron model of Brette and Gerstner [6] as implemented in the NEST simulator [7]. The AEIaF model is a conductance-based integrate and fire model with an exponential soft spiking threshold rather than a hard threshold, and with a second state variable that recreates membrane potential and spike adaptation effects. With this generic model we can characterize neurons showing similar

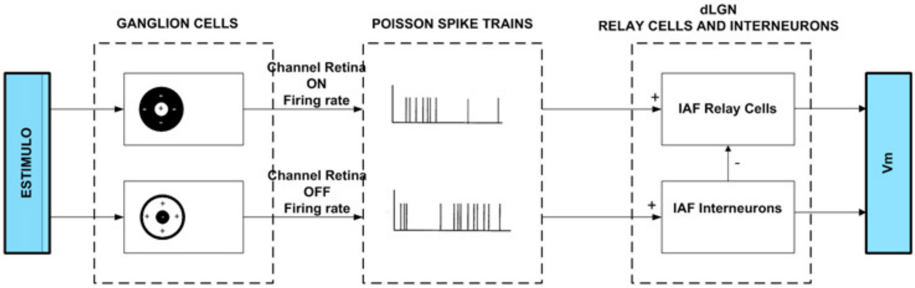


Fig. 2. General overview of the LGN mode. The figure shows components and relations involved in our model, stimuli, receptive field, Poisson generators, IAF neurons.

behaviors to those observed in relay cells and interneurons. The membrane potential is computed according to the following equation:

$$C_m \frac{dV}{dt} = -g_L(V - E_L) + g_s(t)(V - E_s) + g_L \Delta_T e^{\frac{V - V_T}{\Delta_T}} - w + \frac{I_e}{A} \qquad (5)$$

where C_m is the capacitance, g_L is the leak conductance, E_L is the resting potential, and the exponential term creates a soft spiking threshold around V_T with softness determined by Δ_T, g_s, I_e.

The synaptic conductances are shaped by the alpha function (Rall 1967, Jack, Noble and Tsien, 1975) with time constant τ_s:

$$g_s(t) = \overline{g_s}(\frac{t}{\tau_s})e^{\frac{-t}{\tau_s}} \qquad (6)$$

w is an adaptation current with time constant τ_w and sub-threshold adaptation level set by a:

$$\tau_w \frac{dw}{dt} = a(V - E_L) - w \qquad (7)$$

A spike event is triggered when the membrane potential diverges due to the exponential term. In practice, a spike is triggered when V reaches a sufficiently large value such as V_T. When a spike occurs, the membrane potential is reset to Vr and a spike adaptation b is added to the adaptation current w:

$$V \geq V_T \begin{cases} V = V_r \\ w = w + b \end{cases} \qquad (8)$$

4 Results

4.1 Impact of Lateral Inhibition in LGN Processing

In the following experiment we present a horizontal bar stimulus with and without the lateral inhibition coming from the OFF channel (Figure 3). The following results were obtained in our simulation for both situations.

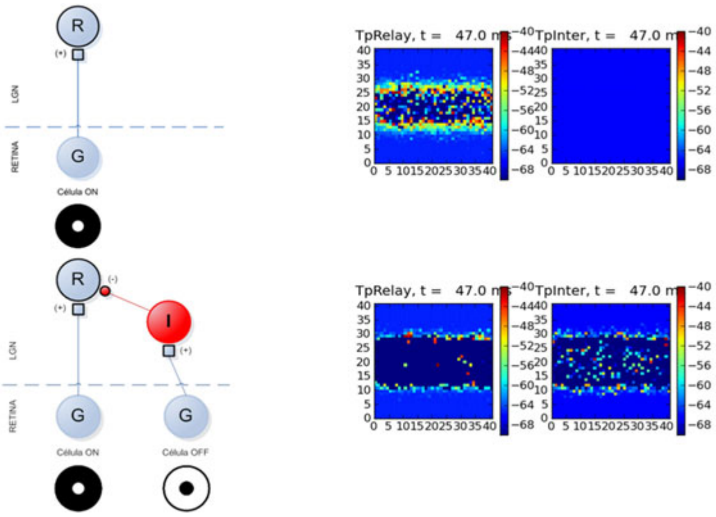


Fig. 3. ON Channel processing and OFF channel inhibition. The figure shows both circuit, with and without inhibition and the difference of activity in the ON relay cell when we have enable and disable the inhibition pathway.

- ON Channel enabled and OFF channel disabled. The simulation shows that at time t = 47 ms the activity of the LGN cell population is centered mainly on the edges of the horizontal bar. Although the ON channel is detecting edges due to the structure of the receptive field DoG, we see that a significant activity remains at its center (see simulation Figure 3).
- ON and OFF Channels enabled. If we present the same stimulus in both channels ON and OFF and we enable the inhibitory pathway coming from the OFF channel, the stimulus edges are detected more precisely (Figure 3). Moreover, the time required to detect the edges is significantly reduced (not shown) when compared with the previous simulation.

It can be concluded that the OFF channel implements a gain control function over the retinal input reaching the ON channel relay cells with the goal of highlighting the contrast of intensity found at the stimulus edges.

4.2 Perception of Stimulus Size Depending on Stimulus Contrast

In the next experiment we have generated different stimulus with varying degree of contrast between the stimulus and background (Figure 4). The results obtained from our simulations show:

- The perceived size of the stimulus in ON channel relay neurons increases as the contrast of the stimulus is increased (Figure 4).

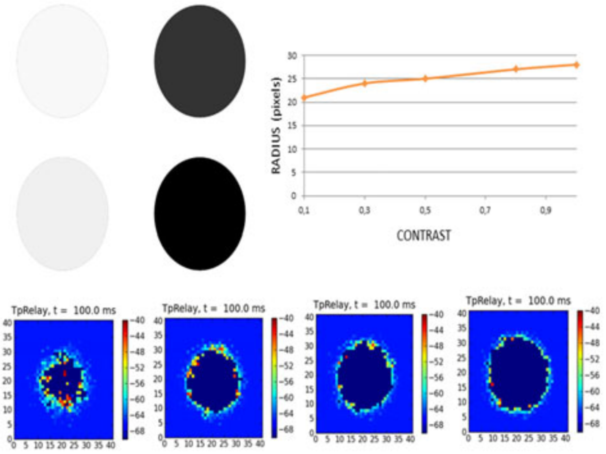


Fig. 4. LGN processing depends on contrast. In the figure we can see different values of contrast in our stimuli and the results for our simulations indicate that there is an increase in the radius of the circular stimulus.

4.3 Impact of Tonic and Bursting Activity in LGN Processing

The thalamic relay cells respond to excitatory inputs in two different modes: tonic and bursting mode. The behavior of both modes depends basically on the calcium current known as I_T.

In the bursting mode, I_T is activated and an influx of calcium causes a depolarization triggering a burst of action potentials. When the relay cell is depolarized for $\sim$ 100ms or more, I_T tends to be de-activated and the cell fires in tonic mode. The aim of this experiment is to analyze what type of filtering is carried out in our LGN circuitry when the relay cell shows a tonic or a bursting activity. Modifying the parameter a in the model AEIaf, we can simulate the behavior of ion channels and therefore change the membrane potential. Values of a $>$ 0 hyperpolarize (potassium current) the membrane, activating the current I_T, and therefore triggering the bursting mode. On the other side, values of a $<$ 0 depolarize (low threshold current of calcium) the membrane, thus inactivating current I_T and therefore generating the tonic mode. The results obtained from our simulations follow:

- When relay cells and interneurons show the tonic mode, we get a better detection of the edges of our stimulus (Figure 5).
- Time detection of the edges of the stimulus is reduced as the membrane potential of relay cells and interneurons becomes more tonic (Figure 5).
- When relay cells and interneurons behave simultaneously in the bursting mode, there is a poor detection of the stimulus edges (Figure 6).
- When relay cells and interneurons behave on modes tonic / bursting and viceversa, the edges are best detected when the stimulus relay cells are working in tonic mode and the interneurons in bursting mode (Figure 6).

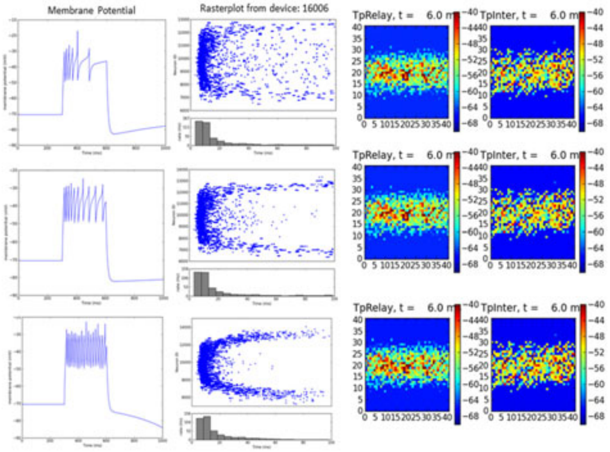


Fig. 5. Impact of tonic activity in NGL processing. Relay cells and interneurons behave tonic mode we get a better detectability of the edges of our stimulus. Time detection of the edges of the stimulus is reduced as the membrane potential of relay cells and interneurons is more tonic.

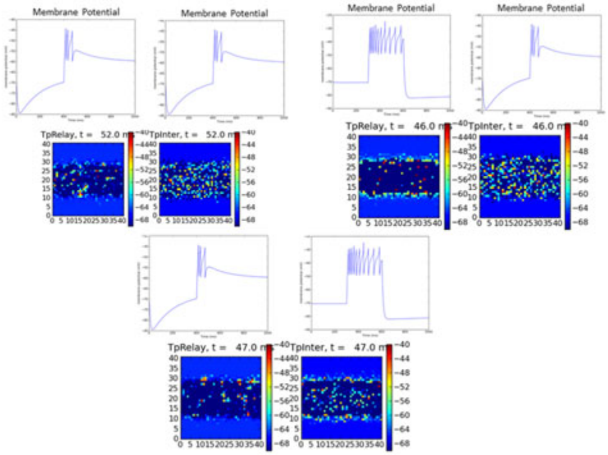


Fig. 6. Impact in bursting activity in NGL processing. Relay cells and interneurons behave simultaneously in burst mode, there is a poor detectability of the stimulus edges. The stimulus edges are best detected when the stimulus relay cells found in tonic mode and interneuron in burst mode.

5 Discussion

Our model of the NGL indicates that this nucleus is probably performing some relevant visual information tasks. Concretely, it provides a plausible explanation of the functionality of the ON and OFF channel push-pull circuit. The center-surround receptive field of the ON channel detects changes in contrast and therefore edges of the presented stimuli. Moreover, we found that the OFF channel (inhibition) is performing a gain control of over the ON channel retinal inputs. The efficacy of the gain control circuit is determined, not only by stimuli features, as for instance the stimulus-background contrast, but also by the activity mode of the LGN cells. Tonic activity in relay cells seems more suitable than bursting activity to detect edges in static stimulus.

The next step will consist on analyzing the difference between the model predictions and the experimental observed activity. To facilitate this future comparison and the relationship between the neural activity and the stimulus features, we are realistically modeling all the ionic channels known in both relay cells and interneurons [8,9].

References

1. Madarász, M., Gerle, J., Hajdu, F., Somogyi, G., Tómból, T.: Quantitative histological studies on the lateral geniculate nucleus in the cat. II. Cell numbers and densities in the several layers. J. Hirnforsch 19(2), 159–164 (1978)
2. Ringach, D.L.: Haphazard Wiring of Simple Receptive Fields and Orientation Columns in Visual Cortex. Journal of Neurophysiology 92, 468–476 (2007)
3. Yeh, C.I., Stoelzel, C.R., Weng, C., Alonso, J.M.: Functional Consequences of Neural Divergence Within the Retinogeniculate Pathway. J. Neurophysiol 101 (2009)
4. Alonso, J.M., Yeh, C.I., Weng, C., Stoelzel, C.: Retinogeniculate connections: a balancing act between connection specificity and receptive field diversity. Progress in Brain Research 154 (2006)
5. Allen, E.A., Freeman, R.D.: Dynamic Spatial Processing Originates in Early Visual Pathways (2006)
6. Brette, R., Gerstner, W.: Adaptive exponential Integrate-and-Fire model as an effective description of neuronal activity. J. Neurophysiol. 94(5), 3637–3642 (2005)
7. Gewaltig, M.O., Diesmann, M.: Nest (neural simulation tool). Scholarpedia 2(4), 1430 (2007)
8. McCormick, D.A., Huguenard, J.: A model of the electrophysiological properties of thalamocortical relay neurons. Journal of Neurophysiology 68, 1384–1400 (1992)
9. Huguenard, J.R., McCormick, D.A.: Voltage clamp simulations of currents involved in rhythmic oscillations in thalamic relay neurones. Journal of Neurophysiology 68, 1373–1383 (1992)
10. Palmer, S.E.: Vision Science. MIT, Cambridge (1999)
11. Dayan, P., Abbott, L.F.: Theorical Neuroscience. MIT, Cambridge (2001)
12. Rieke, F., Warland, D.: Spikes exploring the neural code. MIT (1999)
13. Koch, C.: Biophysics of Computation. Oxford University Press, Oxford (1999)
14. Manzón, M.M., Rodríguez, M.M., Otero, L.M.: DEA. How the thalamus chages What the cat′s eye tells the cat′s brain (2009)

Task-Driven Species in Evolutionary Robotic Teams

P. Trueba, A. Prieto, P. Caamaño, F. Bellas, and R.J. Duro

Integrated Group for Engineering Research,
Universidade da Coruña, 15403, Ferrol, Spain
{pedro.trueba,abprieto,pcsobrino,fran,richard}@udc.es
http://www.gii.udc.es

Abstract. This paper deals with the problem of obtaining coordinated behavior in multirobot systems by evolution. More specifically, we are interested in using a method that allows the emergence of different species if they are required by the task, that is, if specialization provides an advantage in the completion of the task, without the designer having to predefine the best way to solve it. To this end, in this work we have applied a co-evolutionary algorithm called ASiCo (Asynchronous Situated Co-evolution) which is based on an open-ended evolution of the robots in their environment. In this environment the robots are born, mate and die throughout the generations as in an artificial life system. In order to show that ASiCo is capable of obtaining species automatically if they are advantageous, here we apply it to a collective gathering and construction task where homogeneous teams are suboptimal.

Keywords: Multi-robot Systems, Evolutionary Algorithms, Coordination, Collective Intelligence.

1 Previous Work

Using evolution for obtaining coordination in a collective system is not new [4] and different evolutionary algorithms have been developed and applied for years [3][5]. An open issue in this field is that of emergent specialization, that is, the automatic definition of heterogeneous teams in response to a dynamic task that requires individuals performing different sub-tasks in order to accomplish the global one. Specialization can be morphological [12] or behavioural [2]. The first question to answer is which type of task would require specialization. In this sense, there is some agreement amongst researchers that if the task can be naturally decomposed into a set of complementary sub-tasks, then specialization is often beneficial for increasing collective task performance [2]. Examples of these tasks are collective gathering [7], collective communication [6] or multi-agent computer games [5].

In the last decade, coevolutionary approaches have provided the most remarkable results in emergent specialization [13]. Within this line, the authors in [1] conclude that the best approach consists in evolving each robot controller in a

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 138–147, 2011.

different population and evaluating them together. This way, if species are required, the subpopulations will preserve diversity. In this line we must point out Multi-Agent ESP [13], a neuroevolutionaty algorithm that creates n populations for deriving n ANN controllers. Each population consists of u sub-populations, where individual ANNs are constructed by allocating a separate population for each hidden-layer neuron of the network and a number of neuron populations are thus evolved simultaneously. An improved version of Multi-Agent ESP is CONE [8], which includes genotype and behavioral specialization difference metrics to regulate genotypes between and within populations. It has been successfully applied in collective gathering and construction tasks.

The main drawback of these coevolutionary approaches is that all of them have been designed for off-line operation, mainly motivated by their main feature of having separate populations for each robot. The high computational requirements of managing as many populations as robots make them inadequate for real time operation. As a consequence, all the changes in the task and the environment must be included in the simulation because there is no real-time adaptation mechanism. However, due to its demonstrated capabilities for automatic specialization, in this work we will follow a coevolutionary approach too, but avoiding the use of several populations and introducing an intrinsic self-organization capability. The algorithm is presented in the next section.

2 Asynchronous Situated Coevolution

Asynchronous Situated Co-evolution (ASiCo) is an evolutionary algorithm inspired by natural evolution in terms of the use of decentralized and asynchronous open-ended evolution. It differs from other bio-inspired approaches such as genetic algorithms or evolutionary strategies where the selection and evaluation of the individuals is carried out in a centralized manner at regular processing intervals and where there exists an explicit fitness function. Furthermore, unlike in traditional evolutionary algorithms, where each individual represents a solution, in ASiCo a solution is provided by the whole population. To this end, the algorithm performs a coevolutionary process where the individuals are situated, that is, all of them ”live” in a scenario and their interactions, including reproduction, are local and depend on spatial and/or temporal coincidence, making the algorithm intrinsically decentralized. Consequently, ASiCo is highly suitable for distributed and dynamic problems. In this sense, the scenario must be the problem itself instead of the usual representation of it.

ASiCo has been successfully applied to different types of optimization problems, like routing [9], autonomous surveillance [10] and shipping freight problems [11]. In these cases, the tasks were not selected due to their specialization requirements but for being highly relevant distributed problems in engineering. Here, we are more interested in analyzing in depth what happens with a task that presents a clear advantage if coordination is achieved with individuals specialized in different sub-tasks. For a detailed explanation of the algorithm we recommend [10]. Nevertheless, it is necessary to present its basic concepts and operation to better understand the experiments and results.

ASiCo basic elements and operation: there are two types of elements or independent components in an ASiCo scenario: *Individuals* and *Environment components*. Environment components are elements whose parameters are fixed (i.e. rocks or walls). Individuals, on the other hand, are active elements containing parameters that change over time and which may be evolved. They include a state, a set of actuators, some associated to actions that modify their own state parameters (like sensing or updating internal parameters), and others that modify state parameters of other elements in the scenario, and a control system that is going to manage the actions it is executing during operation. The rules that define the interactions among elements and between elements and the scenario are called the *Interaction Set*. Interactions usually imply cost in terms of energy and sometimes may result in an energetic gain. Consequently, the operation of the scenario and of all of the participating components is regulated through the generation and use of *energy*. Thus, one of the main elements in ASiCo is the energy flow strategy, which represents the rules that regulate energy variations and transmission between the individuals and the scenario and vice versa. We have taken inspiration from the studies of utility functions and their distribution among individuals in order to structure the energy dynamics of the environment to guide evolution to the task objective. Specifically, we have used the principled evaluation function selection procedure for evolving coordinated multirobot systems developed by Agogino and Tumer [1], which establishes a formal procedure to obtain the individual utility function from the global function.

The other main element of ASiCo is the reproductive strategy, which is the set of rules that regulates the reproduction process. Reproduction must be defined for each problem and is triggered by spatial interactions together with some energetic and affinity criteria. Within ASiCo, a reproduction mechanism has been designed for fixed size populations called Embryo Based Reproduction (EBR). EBR has been discussed in detail in [10].

Once a problem has been modeled in terms of a scenario and the individuals and elements of this scenario have been defined, the ASiCo algorithm itself starts to operate. The general structure of this algorithm is displayed in Fig. 1 right. It is conceptually divided into two concurrent processes: the interactions defined by the model of the problem and the operators for modifying the structure of the population; this is, creating and deleting individuals during the course of simulation. ASiCo combines in a single process the simulation of the model for continuously evaluating each individual and the evolutionary elements that permit modifying the structure of the population to solve the problem.

Species emergence: one of the main features of ASiCo is self-organization of the individuals into species if required by the task. That is, ASiCo can be seen as an Artificial Life-based system where the individuals that make up the population constitute the solution to the problem while they are in continuous evolution, like in natural systems. This way, when evolution starts, the individuals are random and heterogeneity is maximum but, as generations pass, they start to reproduce and die controlled by a fitness or utility function that affects the whole system. If the solution is unreachable, the population will become

```
Do Initialization:
    For each element ∈ Scenario
        For each parameter ∈ element
            Random generation within parameter range
:End
While Scenario active:
    For each interaction ∈ Interaction set
      if(interaction(i).preconditions == TRUE)
        Interaction(i).modifyparameters
    For each individual
        if(elimination.preconditions == TRUE)
            individual(j).elimination
        if(mating.preconditions == TRUE)
            new individual = individual(j).mating(partner)
:End
```

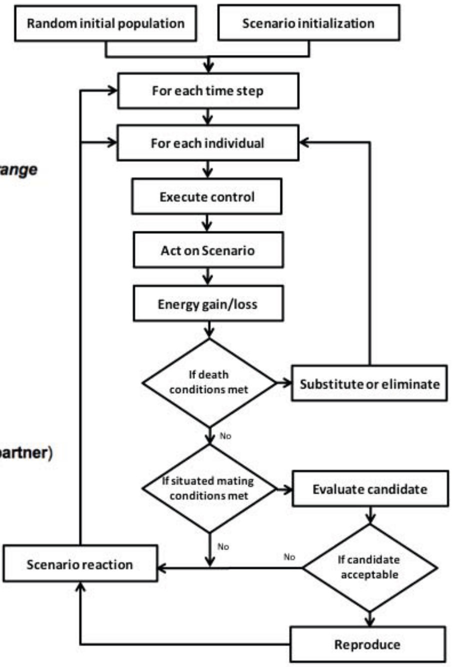


Fig. 1. ASiCo pseudo code (left) and general structure (right)

extinct. In any other case, the individuals will be autonomously organized into species as required by the task, reaching the optimal solution or a suboptimal one. What must be pointed out here is that there is no bias in the algorithm that establishes the heterogeneity of the population, and such a distribution will depend only on the task characteristics, like in a natural evolutionary system. Obviously, homogeneous populations are possible too.

Self-organization is continuous in ASiCo, and once a stable state is achieved, if something changes in the environment or in the task definition, the population will adapt to the new conditions or become extinct. This adaptation implies a very interesting property of this algorithm because distributing the individuals into species implies an autonomous capacity for finding how many are required of each type. Most real problems are dynamic, but to a degree that requires a fast adaptation method in terms of redistributing the individuals between roles or specialties in real time, and not so much in terms of creating new species.

In [10] and [11], heterogeneous populations were obtained with ASiCo but in both cases there was a spatial separation in the scenario between the constitutive elements of the tasks, creating different areas in the environment where individuals could evolve separately, and thus, achieve specialization. In this work, we have defined a collective task where such spatial separation does not exist, and where the individuals are in continuous interaction.

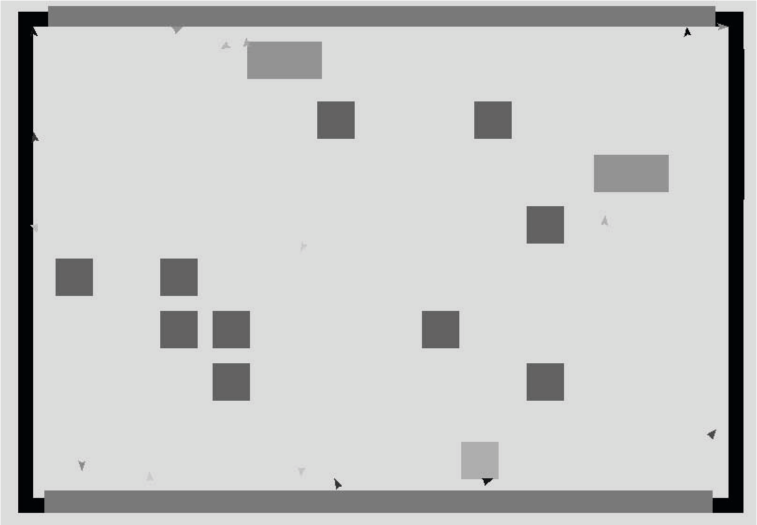


Fig. 2. Representation of the application environment

3 Collective Gathering and Construction Task

The experiment involves a group of robots that must carry out a collective gathering and construction task. The robots must find blocks in the environment, activate them (basically equivalent to taking them out of a box or pallet), push them towards another block and assemble them and, finally, take the assembled units to the collection areas. The *Individuals* and *Environment components* that define the problem have been implemented within ASiCo. The resulting scenario is displayed in Fig. 2 containing the following elements:

The blocks: they represent elements that the robots need to locate and move to carry out the task. They can be in three states: inactivated (dark squares), activated (light squares) or assembled (dark rectangles). An inactive block is an isolated block that has not been found by any robot. Thus, a block becomes active when it is found and acquired by any robot, changing its state. An assembled or built block is created when two isolated blocks are joined.

The collection areas: these are zones of the environment where the assembled blocks must be placed. They are represented by the top and bottom sides in the scenario of Fig. 2.

The robots: they move around the environment looking for blocks to activate, to assemble or to move to the collection zones. They are represented by triangles with a color that varies depending on the robot energy level. Each robot has six sensors that provide the distance and angle to the nearest block of each type. They also have two state parameters, position and age, which vary during lifetime depending on the behavior of the robot. In addition, there are eight descriptive parameters that are fixed for all the robots: vision angle, maximum rotation angle and six coefficients $s_{empty}^{max}, s_{isolated}^{max}, s_{assembled}^{max}, s_{empty}^{min}, s_{isolated}^{min}, s_{assembled}^{min}$, which represent the maximum and minimum speed of the robot when it is not carrying a block (empty) and when it is carrying an isolated or assembled block, respectively. Regarding actuators, each robot has two: one for basic motion and the other second is a gripper for grasping blocks. These elements and parameters are required to describe each robot behavior, but they are not included in the chromosome and, consequently, they are not affected directly by the ASiCo algorithm. In this sense, the robots have the following inheritable parameters:

- Three coefficients, $c_{empty}, c_{isolated}, c_{assembled} \in [0, 1]$, that establish the effective robot speed in the three following situations: exploring the environment, carrying activated blocks or carrying assembled blocks. The effective speed in each case is calculated using equations (1) to (4):

$$c_{norm}^{i} = \frac{c_i}{\sum c_i}, i = empty, isolated, assembled \tag{1}$$

$$s_{assembled} = \frac{1}{\frac{1}{s_{assembled}^{max}} + \left(\frac{1}{s_{assembled}^{min}} - \frac{1}{s_{assembled}^{max}}\right)(1 - c_{norm}^{assembled})} \tag{2}$$

$$s_{isolated} = \frac{1}{\frac{1}{s_{isolated}^{max}} + \left(\frac{1}{s_{isolated}^{min}} - \frac{1}{s_{isolated}^{max}}\right)(1 - c_{norm}^{isolated})} \tag{3}$$

$$s_{empty} = \frac{1}{\frac{1}{s_{empty}^{max}} + \left(\frac{1}{s_{empty}^{min}} - \frac{1}{s_{empty}^{max}}\right)(1 - c_{norm}^{empty})} \tag{4}$$

- Three coefficients that establish the vision range of each type of block, $c_{inactivated}, c_{activated}, c_{assembled\_block}$. The maximum vision range depends on the block size (L) which was fixed in these experiments to 50 pixels. According to this, the maximum distance to detect a block is $2 \cdot L$ pixels of distance for inactivated blocks, $5 \cdot L$ for activated blocks and $7 \cdot L$ for assembled blocks. The effective range of vision ($r_i, i = inactivated, activated, assembled\_block$) for each robot on each situation is calculated through equations (5) to (8):

$$c_{norm}^{i} = \frac{c_i}{\sum c_i}, i = inactivated, activated, assembled\_block \tag{5}$$

$$r_{inactivated} = c_{norm}^{inactivated} \cdot 2 \cdot L \tag{6}$$

$$r_{activated} = c_{norm}^{activated} \cdot 5 \cdot L \tag{7}$$

$$r_{assembled\_block} = c_{norm}^{assembled\_block} \cdot 7 \cdot L \tag{8}$$

- Two grasping parameters h that determine the probability of grasping each type of block (isolated or assembled) . This parameter must be in the range between 0 and 1, and it is governed by the following equation:

$$p_{h=} \begin{cases} 1 & h > 0.66 \\ \frac{h-0.33}{0.33} & 0.33 < h < 0.66 \\ 0 & h < 0.33 \end{cases} \tag{9}$$

- The n coefficients of the control system. Each robot has a control system represented by a RBF ANN with 9 inputs: the distance and angle to the nearest inactivated block, activated block and assembled block, and the robot speeds $s_{assembled}$, $s_{isolated}$ and s_{empty}. The ANN has only one output providing the motion angle of the robot. The number of coefficients n depends on the selected number of ANN hidden layers and neurons per layer.

Once the *Individuals* and *Environment components* have been defined, implementing this experiment in ASiCo implies defining the chromosome representation, the energy flow strategy and the interaction set. As commented above, the chromosome is made up of the n ANN coefficients, the three coefficients that establish the effective robot speed, two grasping parameters and the three vision range parameters. Consequently, the ASiCo evolution has to control both morphological and behavioral parameters of the robots. The energy flow strategy has been defined as follows: each robot is born with 10000 energy units and starts consuming one unit on each time step when it reaches the age of 1000 (representing a mature age). The task is considered to be accomplished every time an assembled block is placed in a collection area. This action provides 2000 energy units to the robot or robots that have participated in the task, that is, to each robot that has found a block, joined two blocks or carried the assembled block to the collection area. When a robot runs out of energy, it dies. As can be seen, the robots increase their private utility when accomplishing the task individually, being the global utility maximization, the one that guides the system towards the completion of the task, a consequence of the maximization of the private one. Finally, creating the ASiCo interaction set implies establishing the rules that define the actions, reactions and perceptions of the robots and the blocks. They are all very simple, and constitute the scenario rules.

The final objective of the multirobot system is simple: collecting the maximum number of assembled blocks (dark rectangles). Hence, previous to the collection of these blocks, the robots must create them by assembling two activated ones (light squares), which must be found by the robots in their inactivated state (dark squares), as displayed in Fig. 2. The task has been designed to allow specialization through the adjustment of the chromosome parameters to exploring the environment, carrying activated blocks or carrying assembled blocks. In this sense, an optimum way to accomplish the task would be using three different species. Obviously, this is not the only option, being the different specialization possibilities the main characteristic of this experiment. Anyway, an homogeneous solution would be suboptimal in this case because the robots that adjust their parameters to intermediate values for the three commented subtasks, will not perform any of them properly.

Experimental results: with the previously presented scenario and elements, ASiCo evolution has been executed several times with a population size of 20 robots during 150000 time steps. The ANN applied had one hidden layer with 9 neurons, implying that 120 coefficients were included in the chromosome to define it. In this section, we are going to present the typical results obtained for

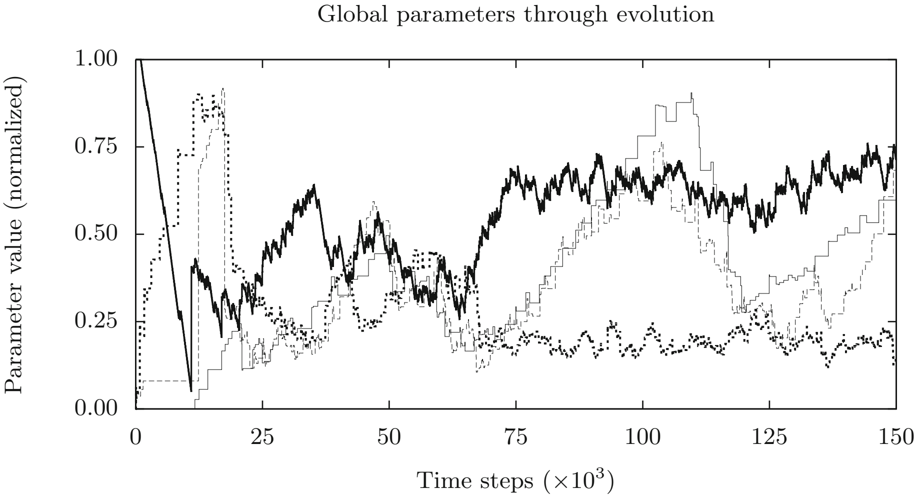


Fig. 3. Evolution of the main parameters that describe the example

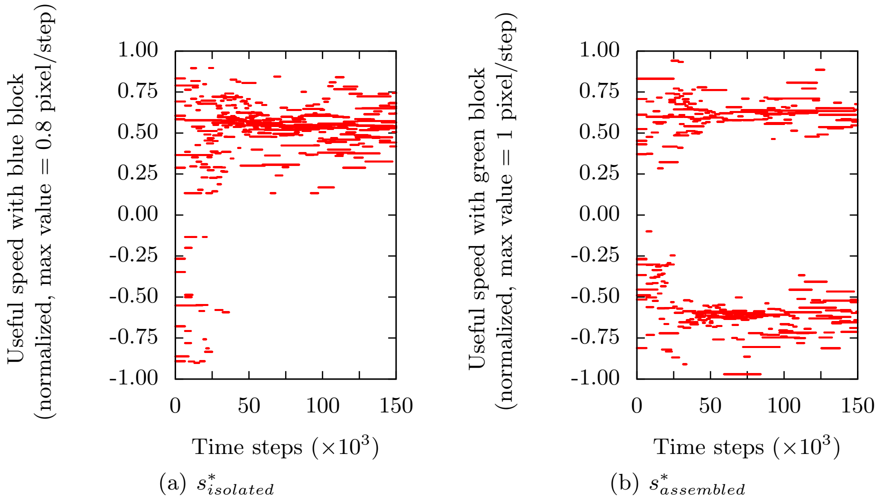
these trials. In Fig. 3 we have represented the evolution of the following parameters: average energy level of the robots (thick line), average task completion time (dotted line), average lifetime (thin line) and average mating selection time (dashed line). The first result that must be pointed out is that the average energy level increases and becomes stable, meaning that the robot team learns to place the assembled blocks in the collection area properly. Furthermore, the average time for task completion clearly decreases and also becomes stable, which is consistent with the energy evolution.

Regarding the other two lines in Fig. 3, and taking into account that the maximum lifetime of a robot is 50000, until time step 15000 the population does not perform the task properly, and most of the individuals die because they run out of energy, this is, they do not reach their maximum lifetime. From this time step forward, the population improves and the robots tend to last until they reach their maximum allowed age (50000). For this reason the graph shows an oscillating behaviour in the population with a 50000 time steps period.

Fig. 4(a) and 4(b) display a representation of the values of the *useful speeds*, $s^*_{isolated}$ and $s^*_{assembled}$, for each robot during evolution. These values are obtained using the following formulas and represent a combination of the speed and grasping capability of the robots when carrying assembled or activated blocks:

$$\begin{aligned} s^*_{assembled} &= s_{assembled} \cdot sign[2 \cdot h_{assembled} - 1] \\ s^*_{isolated} &= s_{isolated} \cdot sign[h_{isolated} - 0.5] \end{aligned}$$

These useful speeds are used to reduce the complexity of the representation of the genetic code of the individuals. Negative values indicate individuals not able



(a) $s^*_{isolated}$

(b) $s^*_{assembled}$

Fig. 4. Useful speed evolution for each robot

(or with low probabilities) to grasp activated or assembled blocks. In Fig. 4(a) we see that during evolution the majority of the robots grasp the activated blocks. Regarding the assembled blocks, we can observe in Fig. 4(b) that there are two clear behaviors, with a portion of robots that grasp them and another portion that ignore them. As a consequence, ASiCo achieves a stable solution where all the robots grasp activated blocks but where only a portion of them is able to grasp assembled blocks too. In addition, the algorithm provides the particular distribution of robots required on each species to solve the task dynamically. This result differs from what we expected, but, in fact solves the problem quite efficiently allowing for all the robots to be able to search the area and assemble blocks and reserving the assembled block transportation capabilities (which is more infrequent, requires more energy and implies less speed) to just some of the robots, which specialize in this task although when idle help out in finding and assembling blocks.

4 Conclusions

In this paper we have applied the Asynchronous Situated Co-evolution (ASiCo) algorithm to a collective gathering and construction task where homogeneous teams are suboptimal in order to show its capabilities for achieving emergent specialization if required by the task. The results confirm that the algorithm obtains two clear species in a scenario where the spatial separation between robots is not an evolutionary advantage, as in previous works. With this specialization, coordination is successfully achieved in an autonomous way.

Acknowledgments. This work was partially funded by the Xunta de Galicia and European Regional Development Funds through projects 09DPI012166PR and 10DPI005CT.

References

1. Agogino, A., Tumer, K.: Efficient evaluation functions for evolving coordination. Evolutionary Computation 16(2), 257–288 (2008)
2. Balch, T.: Measuring robot group diversity. In: Robot teams: From diversity to polymorphism, pp. 93–135 (2002)
3. Baldassarre, G., Nolfi, S., Parisi, D.: Evolving mobile robots able to display collective behavior. Artificial Life 9(1), 255–267 (2003)
4. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, Oxford (1998)
5. Bryant, B., Miikkulainen, R.: Neuro-evolution for adaptive teams. In: Proceedings of the Congress on Evolutionary Computation, pp. 2194–2201 (2003)
6. Floreano, D., Mitri, S., Magnenat, S., Keller, L.: Evolutionary conditions for the emergence of communication in robots. Current Biology 17(1), 514–519 (2007)
7. Li, L., Martinoli, A., Yaser, A.: Learning and measuring specialization in collaborative swarm systems. Adaptive Behavior 12(3), 199–212 (2004)
8. Nitschke, G., Schut, M., Eiben, A.: Collective Neuro-Evolution for Evolving Specialized Sensor Resolutions in a Multi-Rover Task. Evolutionary Intelligence 3(1), 13–29 (2010)
9. Prieto, A., Caamaño, P., Bellas, F., Duro, R.J.: Population Dynamics Analysis in an Agent-based Artificial Life System for Engineering Optimization Problems. In: Proceedings CEC 2009, pp. 2724–2731 (2009)
10. Prieto, A., Becerra, J.A., Bellas, F., Duro, R.J.: Open-ended Evolution as a means to Self-Organize Heterogeneous Multi-Robot Systems in Real Time. Robotics and Autonomous Systems 58, 1282–1291 (2010)
11. Prieto, A., Bellas, F., Caamaño, P., Duro, R.J.: Solving a heterogeneous fleet VRPTW with the asynchronous situated coevolution algorithm. In: Proc. ECAL (2009)
12. Watson, R., Ficici, S., Pollack, J.: Embodied Evolution: Distributing an Evolutionary Algorithm in a population of Robots. Robot. and Auton. Syst. 39(1), 1–18 (2002)
13. Yong, C.H., Miikkulainen, R.: Coevolution of Role-Based Cooperation in Multiagent Systems. IEEE Trans. on Auton. Mental Develop. 1(3), 170–186 (2009)

Concurrent Modular Q-Learning with Local Rewards on Linked Multi-Component Robotic Systems

Borja Fernandez-Gauna, Jose Manuel Lopez-Guede, and Manuel Graña

University of the Basque Country(UPV/EHU)

Abstract. Applying conventional Q-Learning to Multi-Component Robotic Systems (MCRS) increasing the number of components produces an exponential growth of state storage requirements. Modular approaches limit the state size growth to be polynomial on the number of components, allowing more manageable state representation and manipulation. In this article, we advance on previous works on a modular Q-learning approach to learn the distributed control of a Linked MCRS. We have chosen a paradigmatic application of this kind of systems using only local rewards: a set of robots carrying a hose from some initial configuration to a desired goal. The hose dynamics are simplified to be a distance constraint on the robots positions.

1 Introduction

We are working on Linked Multi-component Robotic Systems (L-MCRS) [2,3]. Our previous work deals with the application of distributed control schemes based on consensus techniques to very simple L-MCRS where the links are modeled as springs [6,7]. As an alternative approach to traditional control, we have proposed [5,9] the application of *Q-Learning* algorithm to learn from experience. We have already reported initial works on the application of a modular approach [4,8] which are improved in this paper. Here we move the system to a desired configuration of the robots and the hose.

The *Q-Learning* algorithm belongs to the family of unsupervised *Reinforcement Learning* (RL) methods [11]. It has become very popular because of its good behavior and its simplicity. The algorithm does not require any *a priori* knowledge about the environment and it can be trained on simulated state transitions. A RL learner is assumed to observe discrete states $s \in S$ from the world, choose a discrete action $a \in A$ to be taken following policy $\pi : S \rightarrow A$ and observe new state s'. A previously defined *reward function* immediately maps perceived states to a scalar real *reward* (r) describing how good or desirable is the new state: $r : S \rightarrow \mathbb{R}$. The *reward* is the immediately observed signal qualifying the observed state, the sum of all the rewards observed over time is called the *value*. Knowledge can be acquired from *episodic* (finite tasks) or *continuous* tasks. Once a reward is obtained, an agent can update its previous knowledge

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 148–155, 2011.

about taking action a in state s (described by a real number $Q(s,a)$) using the following expression:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \cdot \max_{a'} \{ Q(s',a') - Q(s,a) \} \right), \quad (1)$$

where $\alpha \in [0,1]$ is the agent's learning speed, $\gamma \in [0,1]$ is a discount-rate parameter that weights earlier rewards higher than later ones. All those *Q-values* are usually stored in a look-up table and each of the entries represents how good an action has been in a state. *Q-Learning* algorithm updates entries as soon as an action is taken based on the estimated value of the new state observed; as it can be seen in equation 1, this estimation is based on the best *Q-value* available in state s'. An action selection algorithm must be defined so an action is selected according to the observed state every time-step. While the straightforward approach (*greedy* action selection) involves always selecting the action with the highest Q-value *exploiting* available knowledge, this prevents the agent *exploring* yet unknown *action-state* pairs and thus, not allowing it to explore possibly better actions. This compromise between *exploration* and *exploitation* is usually solved using a $\epsilon - greedy$ algorithm (a random action is selected with probability ϵ while the best action is chosen with probability $1 - \epsilon$). Episodic tasks are terminated either when a terminal state $s \epsilon S^t$ is reached or when the number of steps reaches a predefined maximum step count.

The main drawback of using *Q-Learning* with explicit look-up tables is called *curse of dimensionality*: as the state space increases, the size needed to store the Q matrix grows so fast it easily becomes an unpractical approach to real applications. To deal with this problem in the L-MCRS control learning, we explore the concurrent learning approach [10] which uses simultaneously multiple RL modules assigning each of them a sub-task to carry out. Each module represents the world as a subspace of the sensorial information available to the agent (ideally, it only receives the state input needed to properly learn its sub-task) and they all are able to learn concurrently. We use local rewards. Regarding the coordination of agents [1], we will assume that robots will receive permission to move in a round-robin schedule, that is, an explicit order will be defined and robots will make discrete moves in turns so robots can consider the remaining robots as static during their turn.

This paper is structured as follows: first we introduce the *Modular Concurrent Q-Learning* concepts and issues in section 2 including features of our own modular approach to L-MCRS. Section 3 presents the application chosen to test our approach. Section 4 shows the results obtained giving details of the conducted experiments. Finally, conclusions and future work comments are presented in section 5.

2 Modular Concurrent Q-Learning

Let n be the number of learning agents in a multi-component system, m the number of modules each agent runs and c the number of possible actions each agent can take. For a discretely perceived world state $s \in S$, the i^{th} module is fed with a subset $s_i \subset s$ which is expected to be relevant to achieve its goal. For the sake of simplicity, all notation will refer to structures and functions existing in all agents and superscripts indicating the agent index will not be added. Let us denote as $Q_i(s_i, a)$ the Q matrix entry of the i^{th} module relating its partial state s_i and action a. Termination conditions are defined as k subsets $S_i^t \epsilon S$ where $i = 1, ..., k$, satisfying $\overset{k}{\underset{j=1}{\cup}} S_j^t = S^t \subset S$.

2.1 Reward and Termination Functions

Most monolithic Q-Learning systems involve the construction of modeling functions that can be expressed as a series of IF-rules: A reward function $r: \ S \rightarrow \mathbb{R}$ and a termination function $t: \ S \rightarrow \{true, \ false\}$:

$$r(s) = \begin{cases} r_1 & if \ s\epsilon S_1^t \\ & ... \\ r_k & if \ s\epsilon S_k^t \\ r_0 & else \end{cases}, \ t(s) = \begin{cases} t_1 & if \ s\epsilon S_1^t \\ & ... \\ t_k & if \ s\epsilon S_k^t \\ false & else \end{cases},$$

where $\overset{k}{\underset{i=1}{\forall}} r_i \epsilon \mathbb{R}$, $\overset{k}{\underset{i=1}{\forall}} t_i \epsilon \{true, \ false\}$, and r_0 represents the selected neutral reward. The modular approach decomposes these functions into m reward functions and m termination functions:

$$r_i(s) = \begin{cases} r_1 & if \ s_i \epsilon S_i^t \\ r_0 & else \end{cases}, \ t_i(s) = \begin{cases} t_1 & if \ s_i \epsilon S_i^t \\ false & else \end{cases}.$$

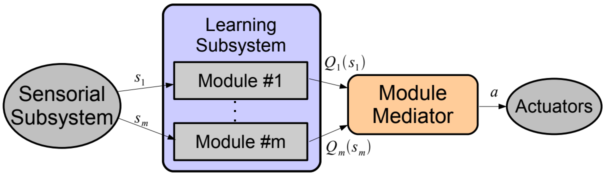


Fig. 1. Modular Q-Learning Scheme

The typical approach to learn different sub-tasks or behaviors concurrently involves using a *Module Mediator* (also referred to as *Module Arbiter* in the literature) responsible for action selection, as represented in Figure 1. Each module has its own Q matrix representing its partial knowledge of the world state s_i and modules may even compete imposing their preferences to the rest. To select the next action we will follow the Greatest Mass (GM) strategy [12] defined as:

$$\pi\left(s_i\right) = \arg\max_{a \in A} \sum_{i=1}^{m} Q_i(s_i, a),$$

which selects the action that maximizes the sum of local agent Q-values.

An additional advantage of using separate modules is that some sub-tasks (those that can be learnt independently of other agents' policies) can be trained separately, thus reducing the exploration space as the state space shrinks to the module's state subspace $\mid s_i \mid$. This reduction often gives the possibility to consider using *exploring starts*[11], which assure a better exploration of the state space.

Two different type of tasks can be distinguished in L-MCRS: those trying to reach one or more goals and those satisfying the physical constraints imposed by the linking element so as to avoid the hard-to-control undesired effects (e.g, force exerted by the hose if stretched). To cope with the different nature of these two module types, we propose distinguishing goal modules and constraint modules while keeping the structure coherent with that already represented in Figure 1. Constraint modules are expected to learn which actions are not to be carried out in a given state, while goal modules learn how to reach the goal not being even aware of the constraints present. We use positive rewards in the goal-oriented modules whenever the goal is reached, negative rewards in the constraint modules if a constraint is not respected and neutral rewards otherwise. The discount-rate parameter γ is different for both types of learning modules: goal-oriented modules will use a rather low γ parameter and constraint-type modules will work best with a high value, so they can learn on a one step basis.

2.2 Veto-Based Action Selection

To enforce the constraints, we propose to use a simple veto system, allowing constraint modules to impose a veto to actions that have broken constraints in the past. A boolean vector $\boldsymbol{V_i} \in \{true, false\}^c$ is defined for each Module Mediator as follows:

$$V_i(a_j) = \begin{cases} true & \text{if } Q_i(s_i, a_j) < v_t \\ false & \text{otherwise} \end{cases},$$

where $i = 1, ..., n$, $j = 1, ..., c$ and v_t is the threshold for imposing the veto. This means that if, under state s_i, a constraint module i has a value $Q_i(s_i, a_j)$ below the veto threshold, action a_j is forbidden for that state. Traditional $\epsilon - greedy$ is then used to select an action among the available ones, assuring this way that taking a random action will not result in an abrupt failure once the modules have a minimum amount of experience, avoiding that way the main problem towards appropriate learning. Instead of manually tweaking the ϵ parameter, we obtain a simple approach to keep goal-focused exploration while avoiding constraint-related termination conditions. Of course, the downside to this technique is that it only works properly with deterministic constraints because exploration is disabled for states found to be undesired in the past.

3 Hose Transport Application

A group of n agents attached at fixed points of a hose must move the hose from an initial state to a goal configuration. The source of the hose is assumed to be located in the $(0,0)$ cell and robots are identified by an integer number in range $[0, n-1]$, where the robot carrying the tip of the hose is labeled as 0 and the rest labels are given according to the position on the hose as shown in Figure 2. In this paper, simple line segments will be used to represent the hose links. Let $P_i = (P_i^x, P_i^y)$ denote the discrete coordinates position of the i^{th} agent on the grid at any time during the simulation. The task consists in reaching a target configuration of the robots $G = \{G_0, G_1, ..., G_{n-1}\}$, with $G_i = (G_i^x, G_i^y)$, starting in the initial hose configuration $I = \{I_0, I_1, ..., I_{n-1}\}$, with $I_i = (I_i^x, I_i^y)$. The hose between i^{th} and $i+1^{th}$ robots is represented as the line segment $P_i - P_{i+1}$, and the hose as a whole has a maximum nominal length of L_{hose} times the size of grid cells. All segments have a maximum length $L = \frac{L_{hose}}{m}$. Because each agent has a different goal: $P_i \rightarrow G_i$, all rewards were local.

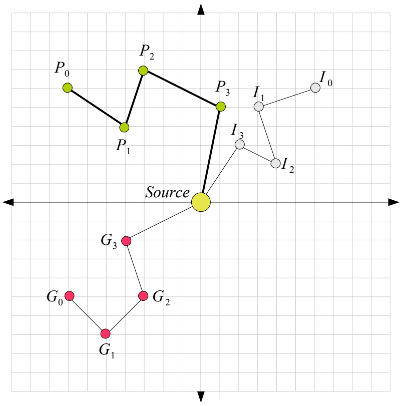


Fig. 2. Representation of the robots and the goal on the grid

3.1 Modules

Our experiments involved a variety of module combinations (including both homogeneous and heterogeneous agents) and the best results were achieved using two goal-modules and three constraint-modules:

1. *Goal-1*: This module models the state as the combination of the distance to the agent's own goal and the angle formed by the hose segment and the goal. Training uses a global reward: whenever the agent reached the goal, a positive reward was given, else a neutral value.
2. *Goal-2*: A simplification modeling the state as the distance to the goal.

3. *Distances*: The module models the length constraint on the hose segment ahead of the robot. It receives a negative reward every time any of them exceeds the maximum allowed length, and a neutral one otherwise.
4. *Collisions*: This module learns to avoid collisions between robots and hose segments. State is modeled as a set of boolean values, each of which indicates whether there is an obstacle (robot or hose segment) within one time-step reach in a direction corresponding to one of the allowed actions.
5. *InGrid:* The robots were desired to stay within a predefined bounds. This module gives a negative reward if the position is outside the allowed bounds, neutral otherwise.

4 Experiments

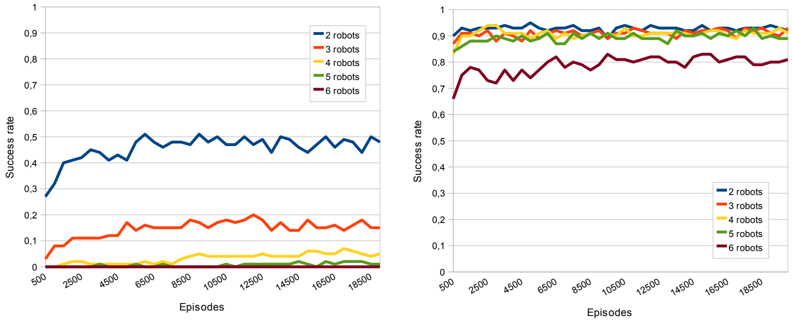
A set of experiments were conducted to test our modular multi-agent learning approach for L-MCRS. We used a grid of 21 × 21 cells, a maximum hose length L_{hose} of 26 cell units and the step limit count for episodes was set to 300: episodes that didn't reach the goal within this limit were forced to abort. In all the cases, the ϵ parameter had a fixed value: 0.1. Robots were allowed to take any of 9 actions at each time step: move North, North-east, East, South-east, South, South-west, West, North-west or no move at all. All of them were considered to be deterministic, always reaching the intended position. Episodes were randomly generated to have a more realistic measure of performance, rejecting those that didn't fulfill all the constraints. We report results on the improvement introduced by the use of the veto system (Experiment B) versus the conventional greedy action selection (Experiment A) in the concurrent training of the agents. Videos were generated for each of the experiments to visually validate the simulations[1]. Figure 3 shows the evolution of the percentage of episodes reaching the target as the training evolves for both experiments for an increasing number of agents. In both cases, increasing the number of agents dramatically reduces the capacity to reach the goal. However results are much better for the veto system (figure 3(b)) than for the baseline greedy system (figure 3(a)).

5 Conclusions

We have presented a veto-based Modular RL techniques suitable to deal with L-MCRS, applying it to the multi-robot hose transport problem. Results show that combined use of separate constraint module training and a veto system leads to a very good learning rate and success rate. We also studied the scalability of the system and, although increasing the number of agents decreased slightly the performance of the learning algorithm, the results obtained are very satisfactory.

In the future, our work will focus on introducing the learnt Q matrices in a simulation environment that includes more realistic hose models for further

[1] Some videos can be downloaded from `http://www.ehu.es/ccwintco/index.php/Borja-videos`



(a) (b)

Fig. 3. Experiment A: Success rate for different number of robots. (a) Experiment A, (b) Experiment B.

validation of the approach and the assumptions made. It also will address the coordination issues so robots can move simultaneously and will consider completely stochastic environments. More complex tasks and scenarios may also benefit from integrating hierarchical structures and this is also a very appealing field of research.

References

1. Busoniu, L., Babuska, R., Schutter, B.D.: A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38(2), 156–172 (2008)
2. Duro, R.J., Graña, M., de Lope, J.: On the potential contributions of hybrid intelligent approaches to multicomponen robotic system development. Information Sciences 180(14), 2635–2648 (2010)
3. Echegoyen, Z., Villaverde, I., Moreno, R., Graña, M., d'Anjou, A.: Linked multicomponent mobile robots: modeling, simulation and control. Robotics and Autonomous Systems 58(12), 1292–1305 (2010)
4. Fernandez-Gauna, B., Graña, M., Lopez-Guede, J.M.: Towards concurrent q-learning on linked multi-component robotic systems. In: HAIS 2011. LNCS (LNAI), Springer, Heidelberg (2011) (in press)
5. Fernandez-Gauna, B., Lopez-Guede, J.M., Zulueta, E., Graña, M.: Learning hose transport control with Q-learning. Neural Network World 20(7), 913–923 (2010)
6. Fernandez-Gauna, B., Lopez-Guede, J.M., Zulueta, E.: Linked multicomponent robotic systems: Basic assessment of linking element dynamical effect. In: Corchado, E., Graña, M., Savio, A. (eds.) Hybrid Artificial Intelligence Systems, Part I, vol. 6076, pp. 73–79. Springer, Heidelberg (2010)
7. Fernandez-Gauna, B., Lopez-Guede, J.M., Zulueta, E., Echegoyen, Z., Graña, M.: Basic results and experiments on robotic multi-agent system for hose deployment and transportation. International Journal of Artificial Intelligence 6(S11), 183–202 (2011)

8. Graña, M., Torrealdea, F.J.: Hierarchically structured systems. European Journal of Operational Research 25, 20–26 (1986)
9. Maravall, D., de Lope, J., Martin, J.A.: Hybridizing evolutionary computation and reinforcement learning for the design of almost universal controllers for autonomous robots. Neurocomputing 72(4-6), 887–894 (2009)
10. Panait, L., Luke, S.: Cooperative multi-agent learning: The state of the art. Autonomous Agents and Multi-Agent Systems 11(3), 387–434 (2005)
11. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
12. Whitehead, S., Karlsson, J., Tenenberg, J.: Learning multiple goal behavior via task decomposition and dynamic policy merging. In: Robot Learning, pp. 45–78. Kluwer Academic Publisher, Dordrecht (1993)

Coordination of Communication in Robot Teams by Reinforcement Learning

Darío Maravall[1], Javier de Lope[1,2], and Raúl Domínguez[1]

[1] Cognitive Robotics Group
Dept. of Artificial Intelligence
Universidad Politécnica de Madrid
[2] Dept. Applied Intelligent Systems
Universidad Politécnica de Madrid
dmaravall@fi.upm.es,javier.delope@upm.es,r.dominguez@alumnos.upm.es

Abstract. In Multi-agent systems, the study of language and communication is an active field of research. In this paper we present the application of Reinforcement Learning (RL) to the self-emergence of a common lexicon in robot teams. By modeling the vocabulary or lexicon of each agent as an association matrix or look-up table that maps the meanings (i.e. the objects encountered by the robots or the states of the environment itself) into symbols or signals we check whether it is possible for the robot team to converge in an autonomous, decentralized way to a common lexicon by means of RL, so that the communication efficiency of the entire robot team is optimal. We have conducted several experiments aimed at testing whether it is possible to converge with RL to an optimal *Saussurean* Communication System. We have organized our experiments alongside two main lines: first, we have investigated the effect of the team size centered on teams of moderated size in the order of 5 and 10 individuals, typical of multi-robot systems. Second, and foremost, we have also investigated the effect of the lexicon size on the convergence results. To analyze the convergence of the robot team we have defined the team's consensus when all the robots (i.e. 100% of the population) share the same association matrix or lexicon. As a general conclusion we have shown that RL allows the convergence to lexicon consensus in a population of autonomous agents.

Keywords: Multi-agent systems, Multi-robot systems, Dynamics of artificial languages, Computational semiotics, Reinforcement learning, Self-collective coordination, Language games, Signaling games.

1 Introduction

In a multi-robot system obtaining a common lexicon or vocabulary is a basic step towards an efficient performance of the whole system [4]. In this paper we present the application of Reinforcement Learning (RL) to the emergence of a common lexicon in a team of autonomous robots. We model the vocabulary or lexicon of each robot as an association matrix or look-up-table that maps the

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 156–164, 2011.

meanings (i.e. the objects, states of the environment and self-states) into *symbols* or *signals*. According to a long and well-established line of thought culminating with the work of Ferdinand de Saussure [9] and Charles S. Peirce [8], the pioneer of Semiotics, the association of the symbols of a language to their meanings are (1) arbitrary and (2) conventional.

In this paper we use arbitrarily (in fact, randomly) initialized association matrices, for each robot and through an dynamic process based on *communicative* or *linguistic interactions* implemented by means of the Reinforcement Learning paradigm, the team converges to an optimum consensus state.[1]

2 Formal Definitions

2.1 Multi-Robot Communication System

We define a Communication System, CS, in a team of robots as the triple:

$$CS \triangleq \langle M, \Sigma, A_i \rangle \tag{1}$$

where $M = \{m_1, \ldots, m_p\}$ is the set of meanings (i.e. the objects or states in the environment that can be of relevance for communication in the team of robots), $\Sigma = \{s_1, \ldots, s_n\}$ is the set of symbols or signals used by the robots in their communication acts and which represent the actual meanings, A_i $(i = 1, \ldots, N)$ are the association matrices of the robots defining their specific associations between meanings and symbols:

$$A_i = (a_{rj})_i \quad ; \quad i = 1, \ldots, N \text{ agents} \tag{2}$$

in which the entries a_{rj} of the matrix A are non negative real numbers such that $0 \leq a_{rj} \leq 1$, $(r = 1, \ldots, p;\ j = 1, \ldots, n)$. These entries a_{rj} give the strength of the association of meaning m_r to symbol s_j; such that $a_{rj} = 0$ indicates no association at all and $a_{rj} = 1$ indicates total association. Note that these quantitative associations have a deterministic and non probabilistic nature so that the associations between meanings and symbols are based on the maximum principle, which means that the maximum value of the entries in a row (column) gives the valid association. An ideal, optimum association matrix is purely binary (the entries are either 0 or 1) and also have the additional restriction of having in each row only one 1 (i.e. no synonyms are allowed) and having a unique 1 in each column, too (no homonyms are allowed).

Optimum association matrices are also known as permutation matrices as they are equal to their transpose matrix, a very important property from the communication efficiency point of view (see paragraph 2.2 below).

[1] Some authors, inspired in the ideas of Ludwig Wittgenstein, have called them *language games* [7,10,2], although we believe a more founded denomination should be *communication* or *signaling games* as defined by David K. Lewis [3].

2.2 Optimal Communication Efficiency

The necessary and sufficient condition for an optimal communication efficiency between two generic robots A_s and A_r is that the transpose of the optimal association matrix of the receiver robot must be equal to the optimal association matrix of the sender robot:

$$A_s = A_r^T \tag{3}$$

where A_s is the optimal association matrix of the robot acting as the sender and A_r is the optimal association matrix of the robot acting as the receiver.

2.3 Optimal Multi-Robot Communication System

Given N robots with respective association matrices: A_1, A_2, $\ldots$ A_N. The necessary and sufficient condition for an optimal global communication efficiency is that all the robots must share the same optimal permutation matrix $A = A_1 = A_2 = \ldots = A_N$.

According to the semiotic tradition, the associations of the symbols of a lexicon to its meanings are arbitrary and conventional. The arbitrariness of the association of a symbol (the signifier in Saussure's jargon) to its meaning (the signified) means that the entries a_{rj} of each robot's association matrix are arbitrarily assigned. The conventional nature of these associations means that all the robots of the population must have the same ideal, optimum permutation association matrix in order to attain an optimum global communication efficiency (we call *Saussurean* such an optimal communication system). This situation is called lexicon or vocabulary consensus and is a hard multi-robot coordination problem. In the sequel we present our experimental work showing that Reinforcement Learning can solve this multi-robot coordination problem.

3 Dynamics of Convergence to Lexical Consensus by Means of Reinforcement Learning

3.1 General Description of the Process

For a team of N robots, each of them with its own lexical matrix A_i (i=1,2,. . . ,N) the way to converge to lexical consensus by means of reinforcement learning is based on the following procedure (see pseudo code in Fig. 1). First, a sequence of what we call language games rounds is performed until the team converges to an optimal communication system or lexical consensus in which all the robots use the same optimal permutation association matrix. In each language game round, all the possible pairwise communication acts are performed (see pseudo code in Fig. 2). Each communication act taking place between two robots with corresponding lexical matrices $\{A_i, A_j\}$ $(i = 1, 2, \ldots, N;\ j = 1, 2, \ldots, N;\ i \neq j)$ proceeds as follows: one of the two robots is selected at random as the speaker (sender) and the other as the listener (receiver). The speaker sends all the possible meanings according to its association matrix. If the meaning decoded by the listener coincides with the speaker's meaning then a success has happened. A failure happens when both meanings differ. After a success the corresponding coefficients of the association matrices in both robots are increased and the competing association coefficients (i.e. a row for the speaker and a column for the listener) are updated in the opposite direction. This additional updating is known as *lateral inhibition* and it is a key element for the convergence process. Similarly, the coefficients involved in a failure are decreased in both robots.

```
for $k = 1, 2, \ldots,$ max rounds do
    Execute all the possible communication acts
    Compute the communicative efficiency of the robot team $EC(k)$
    if $EC(k) = Max$ in three consecutive rounds then
        Break
    end if
end for
```

Fig. 1. Pseudo code of the reinforcement learning-based lexical coordination procedure

```
Assign randomly the sender/receiver roles
for $k = 1, 2, \ldots,$ number of meanings do
    Send the meaning $m_k$ according to the sender's association matrix
    Decode the received symbol $s_k$ according to the receiver's association matrix
    Update both matrices depending on the communication result
end for
```

Fig. 2. Pseudo code of a communication act

The ultimate goal is that after the execution of all the language games rounds the robot team converges to an optimal communication system in which all the robots use the same optimal permutation matrix (optimal *Saussurean* solution).

3.2 Algorithms for the Updating of the Association Matrices

We have applied two different algorithms for the updating of the coefficients of the association matrices: (a) an Ant Colony Optimization-based algorithm, or ACO-like for short, and (b) the incremental algorithm.

In the ACO-like algorithm the coefficients of the association matrix are updated as follows:

$$\begin{gathered} a_{ij}(k+1) = \rho\, a_{ij}(k) + (1-\rho)\,\beta(k) \\ 0 \le \rho \le 1 \\ \beta(k) = \begin{cases} 1 & \text{if reward/success} \\ 0 & \text{if punish/fail} \end{cases} \end{gathered} \tag{4}$$

in which ρ is a critical parameter which has to be carefully selected [1].

In the incremental algorithm [2] the coefficients are updated as follows:

$$
\begin{aligned}
a_{ij}(k+1) &= a_{ij}(k) + \Delta\,\beta(k) \\
0 \leq \Delta &\leq 1 \\
\beta(k) &= \begin{cases} 1 & \text{if reward/success} \\ 0 \text{ or } -1 & \text{if punish/fail} \end{cases}
\end{aligned}
\tag{5}
$$

where the parameter Δ is also critical.

4 Experimental Results

As we are mainly interested in on-line, real-time implementation of lexical coordination in physical multi-robot systems our simulation experimental work has revolved around moderate team sizes typical of multi-robot systems, in the order of 10 or less robots and for them we have experimented with different lexicon dimensions and complexity, as regards speed and stability convergence.

We have conducted series of 600 experiments with the lexicon dimensions from 2×2 to 7×7 (meanings × signals). In all these cases we have centered our efforts on studying the convergence to a common *Saussurean* lexicon for teams of 5 and 10 robots.

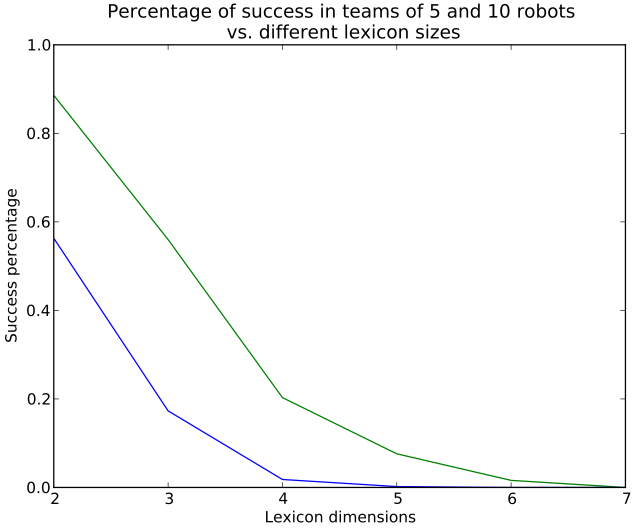
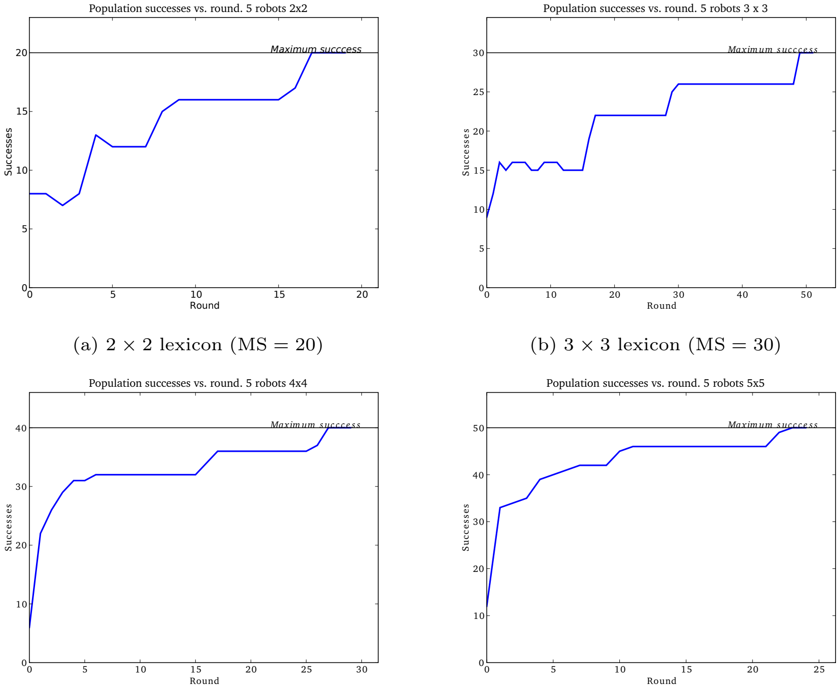


Fig. 3. Rate of success on the consensus to a *Saussurean* communication system for cases of 5 and 10 robots and lexicon dimension from 2 signals and 2 meanings to 7 signals and 7 meanings. Each case of the results presented is based on 600 repetitions and the maximum number of rounds was set to 500. Upper curve: 5 robots. Lower curve: 10 robots.

Once evaluated the convergence dynamics, we conclude that the considered coordination problem can be solved by reinforcement learning techniques with a sufficient success rate for the cases of lexicons of 2 meanings and 2 signals and that the RL method is particularly efficient in the case of teams of 5 robots.

As the lexicon complexity or the team size increases the rate of success of convergence to a *Saussurean* communication system decreases due to the fast growth of the problem's search space. Thus, for 5 robots teams with lexicons of 6 meanings and 6 symbols or higher the cases of complete convergence are more scarce, as it also happens with experiments concerning teams of 10 robots and lexicons from 4 × 4 on. In Fig. 3 the percentage of success for several such cases are presented.

In all these experiments the learning algorithm applied is the ACO. It punishes the association signal-meaning used in both sender and receiver when communication fails. In the case of success, the applied association is reinforced in both communicators and lateral inhibition is also applied in the following manner: the sender's competitive signals and the receiver competitive meanings are punished.



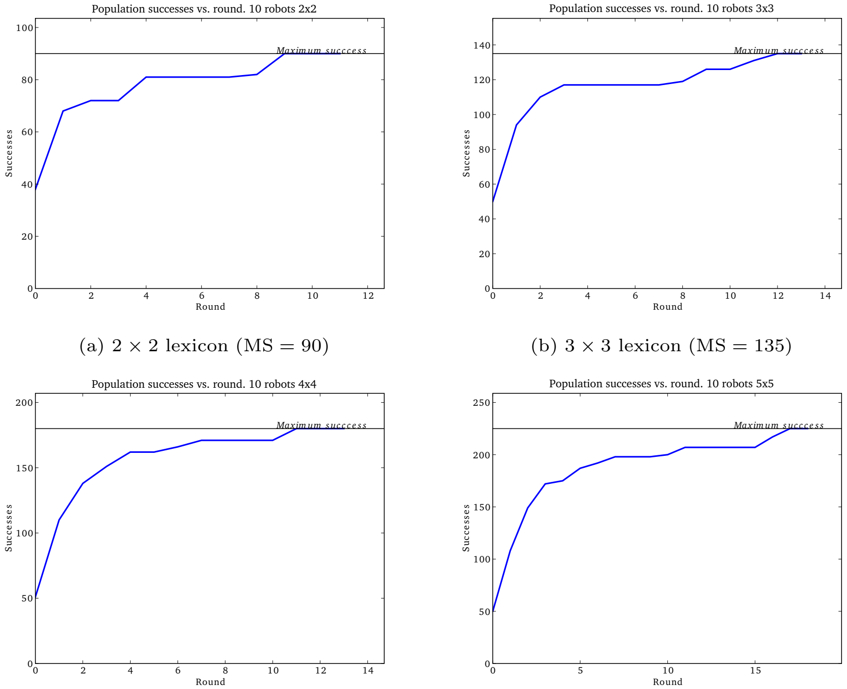
(a) 2 × 2 lexicon (MS = 20)

(b) 3 × 3 lexicon (MS = 30)

(c) 4 × 4 lexicon (MS = 40)

(d) 5 × 5 lexicon (MS = 50)

Fig. 4. Convergence to a *Saussurean* communication system with different lexicon sizes for the case of 5 robots teams. In these graphs the number of round is displayed on the horizontal axis and the vertical axis is for the number of success. Maximum success (MS) is indicated in each graph.



(a) 2 × 2 lexicon (MS = 90)

(b) 3 × 3 lexicon (MS = 135)

(c) 4 × 4 lexicon (MS = 180)

(d) 5 × 5 lexicon (MS = 225)

Fig. 5. Convergence to a *Saussurean* communication system with different lexicon sizes for the case of 10 robots teams. In these graphs the number of round is displayed on the horizontal axis and the vertical axis is for the number of success. Maximum success (MS) is indicated in each graph.

In Fig. 4 and Fig. 5 we present some cases of learning curves displaying the convergence results when consensus is achieved. In all cases the maximum success per round (MS) is calculated using the formula:

$$\mathrm{MS} = \frac{N \cdot (N-1)}{2} \cdot n \qquad (6)$$

where N is the number of robots, and n is the number of meanings. Thus, $\frac{N \cdot (N-1)}{2}$ is the number of encounters and $\frac{N \cdot (N-1)}{2} \cdot n$ is the number of communications per round. Each communication success is counted once and at the end of the round it is verified if all communications were successful (i.e. number of successes is equal to MS). In case this happens for three times we consider that the system has achieved the consensus to a *Saussurean* system.

In the experiments reported in this paper and carried out under the reinforcement learning paradigm, most of the cases of successful convergence to to an optimal *Saussurean* communication system happened before the fiftieth round.

Thus, we can conclude that on average reinforcement techniques provide a faster convergence than the evolutionary strategies used in our previous work [5,6].

5 Conclusions and Further Research Work

In previous work [5,6] we showed that evolutionary algorithms can solve the optimal *Saussurean* communication coordination problem in populations of autonomous agents.

After the experimental results reported in this paper we can conclude that Reinforcement Learning (RL) techniques, besides the advantage of their on-line nature, are efficient enough to solve the communication coordination problem for teams of autonomous robots, and even provide a faster convergence to the optimal solution than evolutionary strategies. It is important to notice that we have focused our experiments on obtaining a strict optimal Saussurean communication system (i.e. we have established a 100% communication consensus in the robot team for considering the convergence as reached, a truly strong restriction).

As we have focused our experiments on the exclusive use of deterministic RL algorithms we plan for the future to experiment also with probabilistic RL algorithms, including the well-known stochastic learning automata methods, to test their efficiency on solving the strict optimal *Saussurean* communication coordination problem.

After our previous experiments with both evolutionary algorithms and with RL methods we can face our next step towards the development and building of a physical multi-robot system based on machine vision for the cognitive part (i.e. for the acquisition and processing of the sensory information related to the meanings of the robots' language) and also based on the use of sound synthesizers for the implementation of the symbols and signals emitted by the robots as their artificial language's words.

References

1. Dorigo, M., Stützle, T.: Ant Colony Optimization. The MIT Press, Cambridge (2004)
2. Lenaerts, T., Jansen, B., Tuyls, K., De Vylder, B.: The evolutionary language game: An orthogonal approach. J. Theor. Biol. 235, 566–582 (2005)
3. Lewis, D.K.: Convention. Harvard University Press (1969)
4. Loula, A., Gudwin, R., El-Hani, C.N., Queiroz, J.: Emergence of self-organized symbol-based communication in artificial cretures. Cognitive Systems Research 11(2), 131–147 (2010)
5. Maravall, D., de Lope, J., Domínguez, R.: Self-emergence of lexicon consensus in a population of autonomous agents by means of evolutionary strategies. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS, vol. 6077, pp. 77–84. Springer, Heidelberg (2010)
6. Maravall, D., de Lope, J., Domínguez, R.: Self-emergence of a common lexicon by evolution in teams of autonomous agents. Neurocomputing (in press)

7. Nowak, M.A.: The evolutionary language game. J. Theor. Biol. 200, 147–162 (1999)
8. Peirce, C.S.: Selected Writings. Dover, New York (1966)
9. de Saussure, F.: Cours de Linguistic Général. Payot, Paris. In: Ibidem Course on General Linguistics, English edn., McGraw-Hill, New York (1916)
10. Steels, L., Kaplan, F.: Bootstrapping grounded word semantics. In: Briscoe, T. (ed.) Linguistic Evolution Through Language Acquisition, pp. 53–73. Cambridge University Press, Cambridge (2002)

Self-organized Multi-agent System for Robot Deployment in Unknown Environments

A. Canedo-Rodriguez[1], R. Iglesias[1], C.V. Regueiro[2], V. Alvarez-Santos[1], and X.M. Pardo[1]

[1] Department of Electronics & Computer Science, Universidade de Santiago de Compostela, Santiago de Compostela, Spain
[2] Department of Electronics and Systems, Universidade da Coruña, A Coruña, Spain
adrian.canedo@usc.es

Abstract. In this paper, we present a multi-agent system for easy and quick robot deployment of robots in different environments, without prior software development and without the need of maps of the environment where the robot will work. Our system, consisting on a network of intelligent cameras and autonomous robots, is able to manage local interactions amongst the intelligent cameras spread across the environment, so that they will self-organize to support robots' navigation and even inform the robots about the existence of events that require their presence. Our design is scalable, robust, and flexible to the environment, so that we get robots able to navigate safely attending events that require their presence. In this paper, we show the performance of a first implementation of our system, which was validated in real world experiments, and upon which we will continue working.

Keywords: Autonomous robots, service robots, sensor networks, intelligent space, multi-agent networks.

1 Introduction

The success of service robots is highly dependant on people's opinion about the services that they provide. On one hand, robots should show some initiative being able to respond to different events on the environment. On the other hand, robots must be flexible, reliable and easy to use: robots need to be able to adapt to different environments and conditions avoiding software or hardware re-design, and they need to ensure continuity of correct service during long periods of time, minimizing their dependence on specialized support. However, the ability of the robots to respond to events is limited due to, amongst other factors, the boundaries of robot's on-board perception. Most of the research that has been done in robotics has focused on self-contained, stand-alone robots that act autonomously. This makes the robot to do all the sensing, deliberation and action selection on board. Nevertheless, the emergence of networked, embedded systems and the increasing presence of networks in homes and factories, opens important possibilities towards the integration of robotic technologies with technologies from the fields of ubiquitous and pervasive computing, sensor networks,

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 165–174, 2011.

and ambient intelligence. This integration provides a new way to build intelligent service robots that opposes the idea of stand-alone robotic platforms. Intelligent sensor networks, which can be spread out over wide areas, will provide an invaluable source of information from beyond robots' immediate surroundings, allowing the robot to respond to a wide range of events, even when they are not taking place next to it. This will make the robot look like it has initiative, and it will improve people's opinion on its role. On the other hand, sensor networks connected to the robot will not only enrich robot's perceptions, but they also could support his movement in the environment and remove or relax the need of detailed maps of the place where the robot moves. Some researchers have been already working on enhancing robots' performance on human-based environments by using intelligent sensor networks. In this sense, some works focused on camera calibration with robots [1], robot localization and motion segmentation [2], robot guidance from cameras over small areas [3], new control architectures aimed towards covering wide areas [4,5,6], and environment mapping supported by camera networks [7]. Finally, some recent works aim at building scene semantic models from visual surveillance networks [8]. Although all these approaches showed interesting results, a scalable, flexible, and robust system for robot navigation has not yet been proposed.

In this paper we describe a multi-agent system, with which we want to achieve a fast deployment of autonomous mobile robots in unknown environments, ensuring robust navigation, and maintaining the cost and the time required for the deployment to a minimum. Our system consists on two kind of agents: a) intelligent cameras spread out over the environment, and b) autonomous robots navigating on it. We based our design on three main requirements: scalability (introducing more agents does not involve hardware or software re-design), robustness (continuity of correct service), and flexibility (independence to the environment or changes on it). In order to get a scalable and robust system, we designed it to be fully distributed, favouring inter-agent interactions, and avoiding any kind of centralization or hierarchy. Moreover, instead of designing global behaviours to fulfil global tasks, our system will be based on low level local interactions amongst agents, leading to self-organized processes, similar to those usual in biology. Finally, our system will not require any kind of pre-defined map of the environment to get intelligent robot behaviours. In the future we plan to use this system to get general-purpose guide robots. Our system will have to negotiate the movement of several robots in different social events at schools, forums, conferences, etc. On the other hand, the robots will have to offer information and advertise the event to people attending or moving around.

In this paper, we present the first prototype of this system. Section 2 gives an overview of the system. Then, Sec. 3 provides a functional description, explaining how the most important duties are achieved. After that, Sec. 4 shows some of the most significant results achieved under real robot operation on real world experiments. Finally, Sec. 5 summarizes our proposal and points out some of our future research.

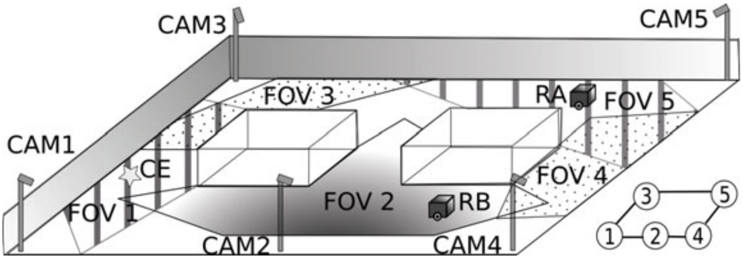


Fig. 1. System's deployment (in this example, camera 1 is detecting an event, robot A is being sighted by camera 5 and robot B by camera 2). We can also see the Field Of View (FOV) of different cameras.

2 System Description

Figure 1 shows a schematic representation of our system. As we have mentioned before, in our system we consider the existence of two kind of agents: *robots* and *cameras* (in this paper the terms *camera-agent* and *camera* are used interchangeably). Robots (RA and RB in the Figure) navigate within the environment avoiding obstacles and moving towards areas where their presence is required (call events from now on, CE in Fig. 1). Cameras, on the contrary, are deployed by the users in the environment, and they have two main functionalities: a) to detect important events within their Fields Of View (FOVs from now on), and to notify these events to the robots if their presence is required, b) to support the robot navigation without a specific map of the environment. As an example of a call event, the camera could detect a group of people to which the robot could offer information or assistance. This ties in with the idea of general-purpose guide robots working in different scenarios.

2.1 Camera-Agents

In this section we describe the hardware and software implementation of the camera-agents.

Hardware Implementation. Camera agents consist on a video camera, a laptop, a WIFI Access Point, several batteries and an aluminium structure built up with a box and a mast attached to it. The video camera is placed on top of the mast, while the bottom box contains the other elements. This design is easy to transport, to deploy and to pick up, allowing us to take our system to different events. Moreover, it does not depend on external power supplies, and does not require any modifications in the environment.

Software Implementation. The software that controls each camera runs four concurrent threads (Fig. 2). We must be aware that this software is the same for every camera:

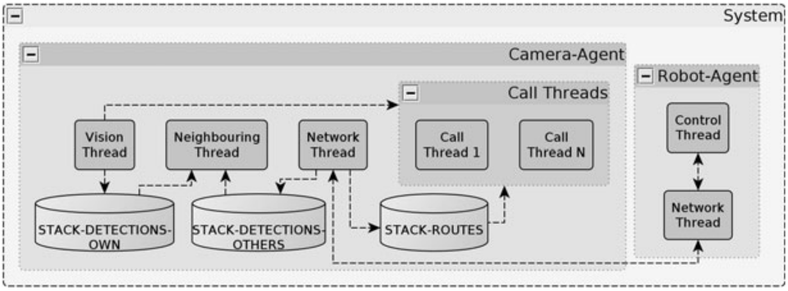


Fig. 2. General software architecture of the system

- *Vision Thread*: This thread is responsible for the detection and tracking of both, robots and call events, in the FOV of the camera.
- *Network Thread*: This thread is responsible for the communications with other agents.
- *Neighborhood Thread*: This thread is responsible for detecting and keeping information about the neighbourhood relationships with other cameras. As we will describe later, each camera will detect which part of its covering area is seen by any other cameras of the system. This allows the dynamic establishment of neighbourhood relationships and local interactions amongst the cameras which, as we will see later, are crucial to get routes, or to help the navigation of the robots.
- *Call Thread*: This thread is responsible for calling the robots whenever there is call event present in the FOV of a camera. It will also negotiate with the robots willing to accept the call, and select one of them. This thread ends either when the robot reaches the event's position or when the event that started the call disappears. There is one of these threads running on the camera for each important event detected in the FOV.

Figures 3, 4, and 5 show a schematic work flow for the Vision, Network, and Call Threads, respectively. These work flows will be further described in Sec. 3.

2.2 Robot

The main goal of the robot is to navigate safely, avoiding obstacles, and attending call events detected from the cameras. We work with wheeled robots such as the Pioneer 3DX, equipped with a laser scanner. The software that controls the robot runs two concurrent threads (Fig. 2): a Network Thread for communications, and a Control Thread for the robot navigation. The navigation is based on the classic but robust Potential Fields method: every obstacle in the local environment around the robot exerts a repulsive force so that the robot avoids any collision, while any goal exerts an attractive force, so that the robot moves towards it. As will see later, this goal and therefore the attractive force is provided from the cameras.

The work flow of the Network Thread is represented in Fig. 6, and a detailed description of it is given in Sec. 3.

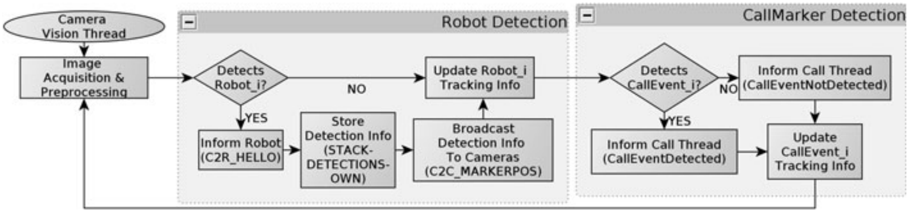

Camera Vision Thread
Image Acquisition & Preprocessing
Robot Detection
Detects Robot_i?
NO
YES
Inform Robot (C2R_HELLO)
Store Detection Info (STACK-DETECTIONS-OWN)
Broadcast Detection Info To Cameras (C2C_MARKERPOS)
Update Robot_i Tracking Info
CallMarker Detection
Detects CallEvent_i?
NO
YES
Inform Call Thread (CallEventNotDetected)
Inform Call Thread (CallEventDetected)
Update CallEvent_i Tracking Info

Fig. 3. Camera's Vision Thread flow chart

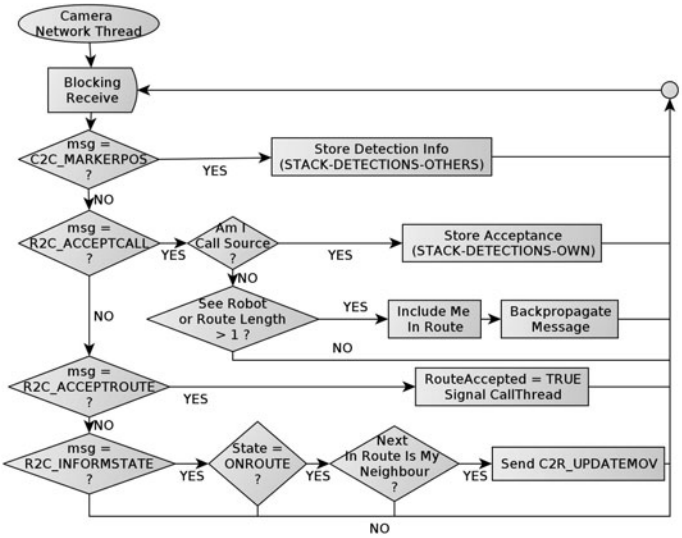


Fig. 4. Camera's Network Thread flow chart

3 Tasks Carried Out by the Camera-Agents

In this section, we will explain the most relevant tasks performed by the cameras to achieve the two main goals described before: dynamic neighbourhood detection, distributed route planning, and support to robot navigation.

3.1 Dynamic Neighbourhood Detection

As we said before, neighbourhood information plays a key role in our system. This information is the basis of all local interactions and self-organization processes. A camera will consider another camera to be its neighbour if they detect the same events simultaneously; therefore this means that the FOV of two neighbouring cameras will necessarily overlap (although we plan to remove this restriction in the future). This overlapping area is called Neighbourhood Region. After a short period of time, all the cameras deployed in the environment

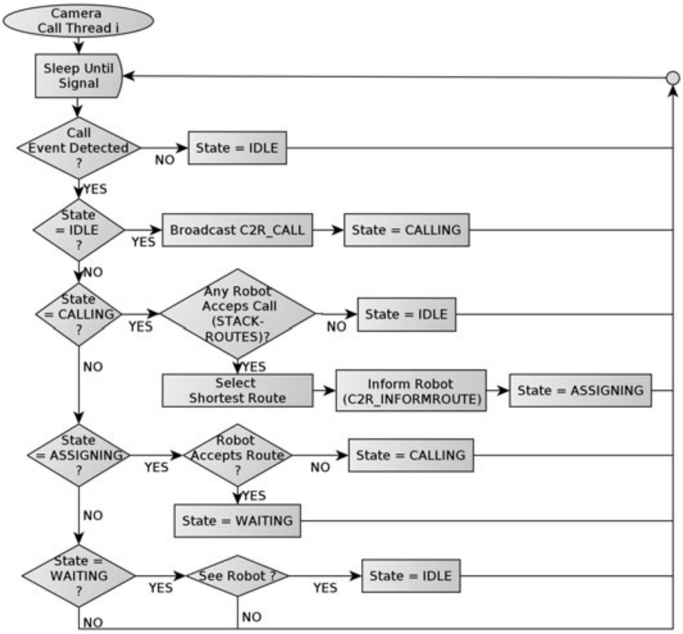


Fig. 5. Camera's Call Thread flow chart

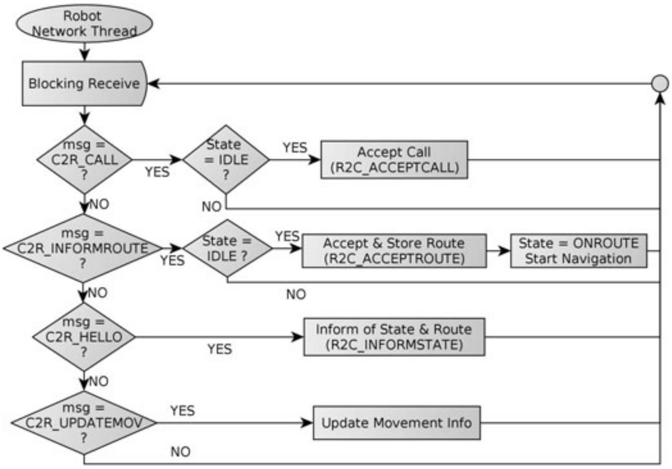


Fig. 6. Robot's Network Thread flow chart

should know their neighbours, and the Neighbourhood Regions amongst them. To get this information we follow the following procedure: as we can see in Fig. 3, whenever the Vision Thread of the camera detects a robot, it stores its position and broadcasts it to all the other cameras (C2R_MARKERPOS).

This information will be received and stored by all cameras (Network Thread, Fig. 4). The Neighbouring Thread running in every camera will periodically check if the detections of the robots are taking place simultaneously from other cameras (Fig. 2), updating the neighbourhood information accordingly.

A very important aspect to notice is that even though this neighbourhood information can change dynamically (e.g. if some camera is moved or stops working), our system will be able to cope with this since this information is updated continuously.

3.2 Distributed Route Planning

A route is an ordered set of cameras through which the robot can navigate: basically, the robot will go from the FOV of one the cameras to the FOV of the next camera on the route. These routes will allow our robots to navigate without metric maps of the environment. Following with our system's philosophy, these routes will be generated on demand as a result of local interactions amongst the cameras, without the intervention of any central entity.

The route formation starts with the detection of a call event that requires the presence of a robot. When this happens, the Vision Thread of the camera detecting it (Fig. 3, CallMarkerDetection) informs the corresponding Call Thread, which in turn will broadcast a request to all the robots (C2R_CALL broadcast, Fig. 5). Upon reception of this message (Fig. 6), every free robot (IDLE state) accepts the call, and broadcasts this acceptance to every camera (C2R_ACCEPTCALL). Then, those cameras which receive the acceptance message and which are seeing the robot which issued that message, will start a back-propagation process (Fig. 4) to create a route; through this process there will be a passing message that will contain a route to go from the FOV where the robot is seen, until the FOV where the event that started the call is. Each camera passing the message will include itself in the route, and forward it to its neighbours (except those that are already included in the route). This process ends when the message is received by the camera which triggered the call. Therefore, this camera should receive information about which robots accept the call and the routes they would have to follow in order to reach its FOV.

The Call Thread of the calling camera (Fig. 5) will select the shortest route received, inform the corresponding robot accordingly (C2R_INFORMROUTE), and wait until the robot arrives to the call event area, provided that the robot has previously accepted the route.

As an example of route formation, consider the hypothetical topology in Fig. 1. Consider that the robot A is free (willing to accept any call), while the robot B is not. In this scenario, when the camera 1 detects the call event (CE) and asks for robots to attend, RA is the only one accepting the call. Then, the camera 5 receives the acceptance, and forwards it to cameras 3 and 4 (its neighbours), which will forward it as well to their own neighbours, and so on. Finally, camera 1 will receive two acceptances with the routes 5-3-1 and 5-4-2-1 coming along, from which the first one will be chosen. It is clear, after this description, that the route formation does not emerge from a globally coordinated process, but from multiple local interactions among agents.

3.3 Support to Robot Navigation

When a robot is following a route, each camera on the route will support the movement of the robot to the FOV of the next camera on the route. We must bear in mind that every camera keeps information not only about the neighbouring cameras but also regarding the neighbourhood regions (overlapping FOV areas). First of all, the camera which sees the robot will start a communication with it (C2R_HELLO in Fig. 3). If the robot is attending a call event (state ON-ROUTE), it will answer to the camera with its state and the route it is following (C2R_INFORMSTATE in Fig.6). Finally, the camera receiving this message estimates the direction that the robot should follow to reach the FOV of the next camera on the route (basically the robot should move towards the Neighbourhood Region shared with the next camera on route), and sends this information to the robot (C2R_UPDATEMOV, in Fig. 3). Once the robot receives information about the direction it should follow, it will consider this direction as an attractive force (Potential Fields based navigation method described in Sec. 2.2). In this manner, the robot will move from the FOV of one camera to the other on the route.

4 Implementation and Experimental Results

We have implemented both kinds of agents: robots and cameras. The robot used was a Pioneer P3DX with a SICK-LMS200 laser. On the other hand, each camera agent used either an Unibrain Fire-i camera, or a PointGrey Chameleon CMLN-13S2C with a FUJINON FUJIFILM Vari-focal CCTV Lens (omnidirectional). The processing units where either a DELL Latitude E550 (Intel(R) Core(TM) 2 Duo P8600 @ 2.4 GHz, 4 GB RAM) or a Toshiba Satellite A100-497 (Intel(R) Core(TM) 2 Duo T5600 @ 1.83 GHz, 4 GB RAM). Regarding the software of the controllers, it was implemented using the Player(v-2.0.3)-Stage(v.2.0.4) platform for the robot, and the OpenCV 2.0 library for the camera agent. Finally, messages were passed over an IEEE 802.11g local wireless network via UDP.

4.1 Experimental Setup and Results

We deployed a multi-agent network over the Department of Electronics and Computing of the University of Santiago of Compostela. As represented in Fig. 7, the network consisted on a robot-agent (R), four camera-agents (A, B, C, D) and a call event (M). D sighted the call event (M), started the robot call process, and triggered the route formation (see Sec. 3.2). Upon reception of the route, R started navigating through the network towards M, supported by A, B, C, and D (in this order, from left to right, top to bottom in the Figure), while avoiding the obstacles detected. The robot's trajectory and the map of the

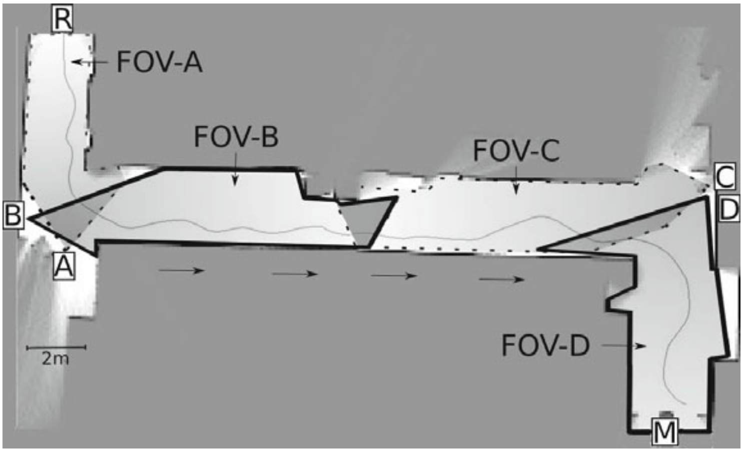


Fig. 7. Trajectory described by the robot in one of the experiments. The robot R navigates from the FOV of camera A, passing through B, C and finally D, which triggered the call upon detection of the marker M.

scenario was obtained from the robot's odometry and laser logs using the PMAP SLAM library, compatible with Player-Stage. We run several experiments changing the initial positions of the robot or the call event, obtaining in all of them a satisfactory performance of the robot.

5 Summary, Conclussions, and Future Work

In this paper, we proposed an initial approach for a distributed and intelligent system for robots deployment over unknown environments, consisting on a network of intelligent cameras and autonomous robots. Our purpose was to enable robots to attend world events happening out of their surroundings, thus increasing robot's initiative. We designed our system on the base of scalability, robustness, and flexibility, independent of the environment and of the number and layout of the agents deployed. In this regard, we did not use any centralization or other kind of hierarchy, but biologically inspired self-organization processes, based on distribution, inter-agent independence, and emergent behaviours out of local interactions, instead of global plans. This resulted on a system highly independent of environment changes, redundant, and flexible enough to cope with a wide range of spatial distributions of the cameras. Also, our system is able to work with several robots or to attend to simultaneous events.

Real world experiments proofed that our system makes it possible for the robot to attend events detected by the cameras, regardless of both robots' and events' positions. Upon distribution in the environment, the cameras detect automatically their neighbours, forming a global camera network out of local neighbourhood relationships. The routes are planned by a distributed process of local interactions, allowing the robot to navigate between each pair of cameras, counting on camera's support.

The work presented here is just the begging of a bigger project. In future stages of our research, we will include the detection of real world events requiring robot's presence, such as groups of people for guiding robots at museums. Also, we will automatize the detection of overlap FOVs attending at people moving within the environment, and even remove the FOVs' overlap restriction at more advances stages. Finally, we plan to improve the robustness of the robot navigation, by learning from people trajectories, by using SLAM techniques, and by utilizing WIFI localization systems.

Acknowledgements

This work was supported by the research grants TIN2009-07737, INCITE08PXIB262202PR, and the BES-2010-040813 FPI-MICINN research grant.

References

1. Rabinowitz, P.: Self-calibration of environmental camera for mobile robot navigation. Robotics and Autonomous Systems 55(3), 177–190 (2007)
2. Losada, C., Mazo, M., Palazuelos, S., Pizarro, D., Marrón, M.: Multi-Camera Sensor System for 3D Segmentation and Localization of Multiple Mobile Robots. Sensors 10(4), 3261–3278 (2010)
3. Fernández, I., Mazo, M., Lázaro, J.L., Pizarro, D., Santiso, E., Martín, P., Losada, C.: Guidance of a mobile robot using an array of static cameras located in the environment. Autonomous Robots 23(4), 305–324 (2007)
4. Lee, J.H., Hashimoto, H.: Controlling mobile robots in distributed intelligent sensor network. IEEE Transactions on Industrial Electronics 50(5), 890–902 (2003)
5. Lee, J.H., Morioka, K., Ando, N., Hashimoto, H.: whereas we used Cooperation of distributed intelligent sensors in intelligent environment. IEEE/ASME Transactions on Mechatronics 9(3), 535–543 (2004)
6. Steinhaus, P., Strand, M., Dillmann, R.: Autonomous robot navigation in human-centered environments based on 3D data fusion. EURASIP Journal on Applied Signal Processing 2007(1), 224 (2007)
7. Rekleitis, I., Meger, D., Dudek, G.: Simultaneous planning, localization, and mapping in a camera sensor network. Robotics and Autonomous Systems 54(11), 921–932 (2006)
8. Makris, D., Ellis, T.: Learning semantic scene models from observing activity in visual surveillance. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35(3), 397–408 (2005)

Selective Method Based on Auctions for Map Inspection by Robotic Teams

Manuel Martín-Ortiz[1], Juan Pereda[1], Javier de Lope[2], and Féliz de la Paz[3]

[1] ITRB Labs
Research, Technology Development and Innovation, S.L
[2] Computational Cognitive Robotics
Universidad Politécnica de Madrid
[3] Dept. Artificial Intelligence
UNED
{manuel.martin,juan.pereda}@itrblabs.eu, javier.delope@upm.es,
delapaz@dia.uned.es

Abstract. In the inspection of a known environment by a team of robots, communication problems may exists between members of the team, even, due to the hostile environment these members can be damaged. In this paper, a redundant, robust and fault tolerant method to cover a known environment using a multi-agent system and where the communications are not guaranteed is presented. Through a simple auction system for cooperation and coordination, the aim of this method is to provide an effective way to solve communication or hardware failures problems in the inspection task of a known environment. We have conducted several experiments in order to verify and validate the proposed approach. The results are commented and compared to other methods.

1 Introduction

Nowadays, the autonomous robotics field covers a wide range of projects, from UAV (Unmanned Aerial Vehicles) [6,10,3] to humanoids with social skills [14,2]. This wide range of works shows the great progress that is been achieved in the last decade and opens a promising path for future research. The main topic of the autonomous robotics focuses on techniques that allow robots to collaborate between themselves in order to achieve a specific task [8] as it can be the exploration of surroundings [4,11]. Multi-robot systems and collaborative robotics [7] are becoming one of the biggest and most exciting challenges in this field.

One of the best applications that could be find in collaborative robotics, is the assistance to human beings in hostile environments. Pipes, sewers or abandoned mines are examples of this type of hostile environments. Projects like Makro [15] or MOIRA [13] and authors like Thrun [18] have already dealt with these environments, marking the lines to follow and emphasizing the importance of inspection tasks in these places. Kawaguchi [9] and Zhang [19] design robotic systems for pipe inspection.

The inspection task can be dealt with solutions based on A* algorithm, like Learning Real-Time A* (LRTA*) [16]. This algorithm used in robot teams needs

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 175–184, 2011.

a centralized server for planning the inspection routes. Likewise, a stable and reliable communications system is needed.

One of the handicaps in this type of underground environments is the difficulty to keep continuous communication among the members that conform the team. This requires (1) high degree of autonomy for each member of the group and (2) all the members must share all the possible information when the environment allows the communication between themselves. Despite new techniques are growing [1,5], empirical evaluations [17] shown that, for example, the soil composite or oil present are a decisive aspect.

In this paper, we introduce a redundant, robust and fault-tolerant method, which is able to perform inspection tasks on environments described above. Inspection tasks assume knowledge about the environment. We model this knowledge like a connected graph to represent the infrastructure of the sewer or pipe network. The agents of the team compete against each other through auctions to prevail the best travel routes and then complete the main task.

The paper is structured as follow. First, we describe how the proposed method works and the requirements that must fulfilled. Second, we expose the different experiments in a brief comparative with other methods. Finally, we comment the conclusions and possible improvements in our method as well as some proposals for future works.

2 Description of the Proposed Method

2.1 Objective

The aim of the proposed method is to inspect, in a finite time, a pipe or sewer environment modeled as a graph by the collaboration of a team of agents. We consider the inspection task as the action to visit each of the edge that comprise the graph by some of the team's members of the multi-robot team. To accomplish this, agents can make use of communications for coordination with the rest of the team when the circumstances allow it. The method proposed is able to overcome failures in the communication or in the robots' hardware. In the worst case, a single agent should be able to, eventually, visit all edges of the graph.

2.2 Prerequisites

Modeling a pipe network or a sewerage system like a graph allows us to work with powerful data structures and make use of graph theory. Thus, nodes represent intersections and edges represent corridors or tunnels.

When an agent reaches a node, it knows which node is. If an agent wants to go from the node A to the node B, eventually it will arrive at node B directly from node A, i.e. an agent placed in the node A knows which edge has to be selected to travel to the node B.

One characteristic of our method is that each agent could start to work at any moment, it is not needed that the agents start altogether. Furthermore, it is possible to lose agents during the inspection task.

2.3 Elements and Structures

In the sequel we enumerate the structures and elements used in our method:

- Graph: The graph is the representation of the real environment where nodes represent intersections and edges represent corridors or tunnels. Isolated nodes are not present in the graph neither small groups of disassociates nodes or islands. Thus, we are working with connected graphs.
- Route: It is a *path* of the graph, where a *path* is *"a sequence of arcs where all the arcs are directed in the same way, i.e. the end of an arc coincides with the origin of the following one"* [12]. The first node of a path always is the agent's current node. The routes represent a set of nodes (and therefore a set of edges) that the agent decides to visit.
- Agent: An agent is the representation for a robot, it will select the routes to be visited. The agent will negotiate the routes with other agents in an auction through a central server.
- Central Server: The aim of this element is to be a shared memory. When agents can communicate with the central server, they will be able to know which edges have been visited for other agents. Likewise, this will be the place where agents bid for routes during the negotiation.

2.4 How It Works

As each member of the team as the central server work independently. An individual member of the team negotiates with other teammates through the central server to accomplish the main task: inspect all edges of the graph. The central server works as a shared memory that allow the team to share knowledge about the graph state and make auctions to make the appropriate decisions.

The following sections describe in more detail the role played by each element.

Central Server. This component has two main tasks: it develops the role of shared memory for all agents and it is also a place where the agents perform the auctions. As shared memory provides to the agents the necessary information to keep their own maps updated with the progress on the global task.

When an agent wants to start an auction, it sends a message to the central server for starting the auction process. Others agents can submit theirs own routes and when finish the time for the auction, the central server proceed to make an election and communicate to each agent if its route is the winner. An agent with a winner route could do the trip through the nodes of the route.

The routes are compared in function of the shared stretchs (edges between two nodes). In terms of set theory, if each route is a set of stretchs to be cover, the central server compares all routes that intersect. The routes that not intersect with any other are selected as winners (see eq. 1). For the routes that intersect with others, the central server will choose as the winner to the route with more exclusive stretchs (see eq. 2), or a random of them in case of equality.

Thus, if Σ is the set of proposed routes by agents, the route a is declared winner in an auction if and only if one of the following equations is satisfied:

$$\exists a \epsilon \Sigma,\ \forall b \epsilon \Sigma,\ a \neq b \wedge (a \cap b) = \phi \tag{1}$$

$$\exists\, a \epsilon \Sigma,\ \ \forall b \epsilon \Sigma,\ \ card(a - C_a) \geq card(b - C_b) \tag{2}$$

where

$$C_a \ = \cup\, c \epsilon \Sigma,\ c \neq a \tag{3}$$

$$C_b \ = \cup\, c \epsilon \Sigma,\ c \neq b \tag{4}$$

Fig. 1 shows an example of route selection in terms of set theory. Each proposed route is denoted with a letter (*A, B, C,...*), each route is composed by stretchs denoted with numbers (*1,2,3,...*). In the situation showed in this figure, routes *A* and *E* will be winners and agents that proposed routes *B, C* and *D* will have to propose a new routes and start a new auction process.

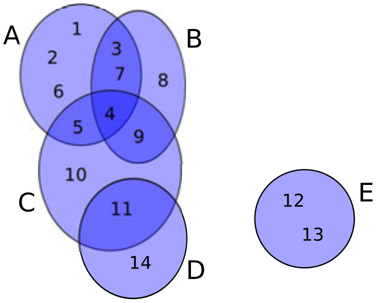


Fig. 1. Route selection process

Agent. The aim of the agents is to achieve that all the edges of the graph will be visited. To accomplish this objective, all members of the team negotiate the routes that best fit to reduce the number of times an edge is inspected.

Throughout the execution of its algorithm (described in Fig. 2), the agent checks if it has a precalculated route. If the route does not exists, then it will calculate the best fit route to its current position. This route selection process is described later.

If a route exists, then the agent checks if this route is confirmed. We employ the term *confirmed route* to call the routes which have been won by the agents in a previous negotiation with other teammates. Normally, these routes can be executed by the robots without the need of communicate with the central server.

If the route is not confirmed and communication with central server is possible, then an auction process will start. In this auction all routes presented compete. If an agent wins an auction, its route is confirmed and it can execute it. Agents with not winner routes should calculate its best next route and start a new auction process.

On the other hand, if the route is not confirmed and the communication is not possible, the agent will execute the first step of its personal trip, trying to contact with the central server in the next node to negotiate this route.

The route selection process is as follow. The agent selects all possible paths starting from its current position on the map. The agent only considers paths with a length of three nodes. This arbitrary number is empirically determined and can be tunned according to size and topology of the map. The criterium followed to establish the length of the paths is to avoid combinatorial explosion in the route search process. Then, preselected routes are evaluated according to the edges: a visited edge is more interesting that a not visited edge. Finally, the selected route for auction process is the most valued route between all the preselected. In case of equality, the agent selects a random path.

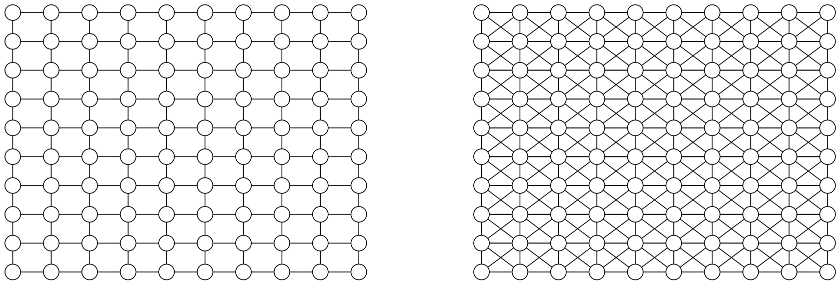
```
route := []
route_confirmed := False
while not is_all_map_inspected():
    if route != []:
        if route_confirmed:
            go_to_next_node(route)
        else:
            if is_communication_possible():
                auction_result := start_auction(route)
                route_confirmed := auction_result
            else:
                go_to_next_node(route)
    else:
        route := select_best_route_from_current_position()
```

Fig. 2. Pseudo-code for robot-agent

3 Experimental Results

We have conducted several experiments to check the worth of the proposed method. The results are compared to the obtained by teams of robots with a random behavior. The criterium followed by these teams is completely random, i.e. each agent randomly selects a not visited edge from its current position and when if all the edges are been visited, a random edge is selected.

The objective to achieve is that all edges of the graph will be visited with the least number of steps. We assume a *step* as the action of *"go from a node A to a node B"*. We will show the results of the experiments for two kind of teams formed by agents, each with a different behavior (one for random behavior and the other with our method). Each behavior is executed by an incremental group of agents, starting with one agent and finishing with teams of thirty agents working together to achieve the main objective. The communication failure probabilities ranges between 0.9 and 0.00001. A total number of 100 tests are carried out for each particular experiment.



(a) Map denominated as *Map 1*

(b) Map denominated as *Map 2*

Fig. 3. Maps designed for experiments

Using a map as the showed in Fig. 3 (a) (here in after *Map1*) with a high and a low probability of communication failure (exactly 0.9 and 0.00001 respectively) were collected the results showed in Fig. 4. Grey bars show the steps needed by the different groups of agents executing a random behavior, black bars are the steps needed by the groups executing the selective method.

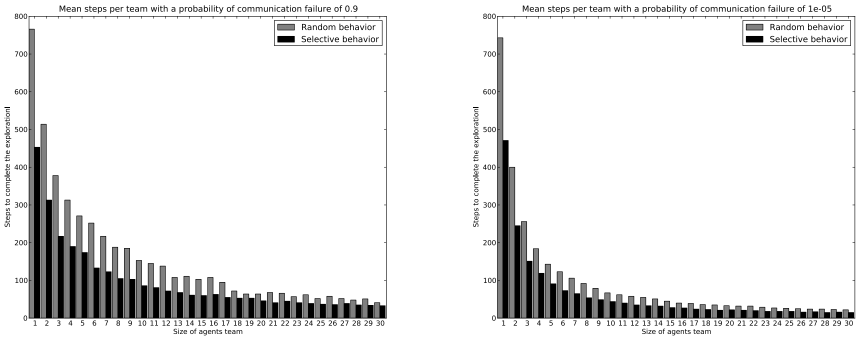
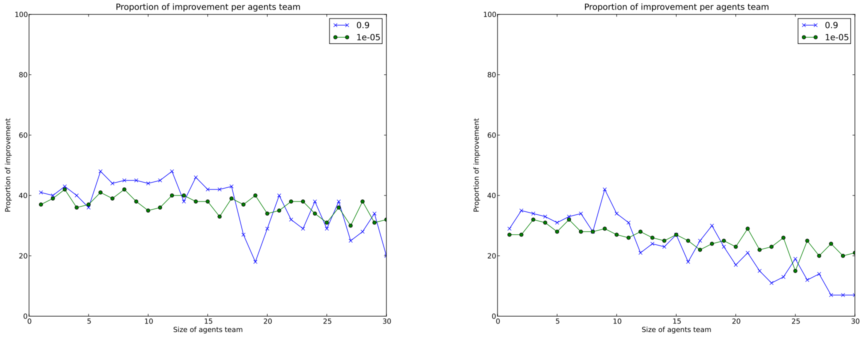


Fig. 4. Experiments results relative to mean steps

Fig. 5 (a) depicts the curve that represent the improvement of our method over the random behavior for both probabilities shown previously. The range of this improvement goes from 50% to 30% independently the size of agent team.



(a) *Map 1*

(b) *Map 2*

Fig. 5. Improvement of our method over the random behavior

The same experiments are performed over the map as showed in Fig. 3 (b) (referenced as *Map2*). This map, with a high connectivity in all nodes, presents a great number of routes on which to choose as best candidate. Fig. 6 shows the results of the inspection task in the *Map 2* by the random and the selective behaviors with a high and a low probability of communication failure (0.9 and 0.0001 respectively).

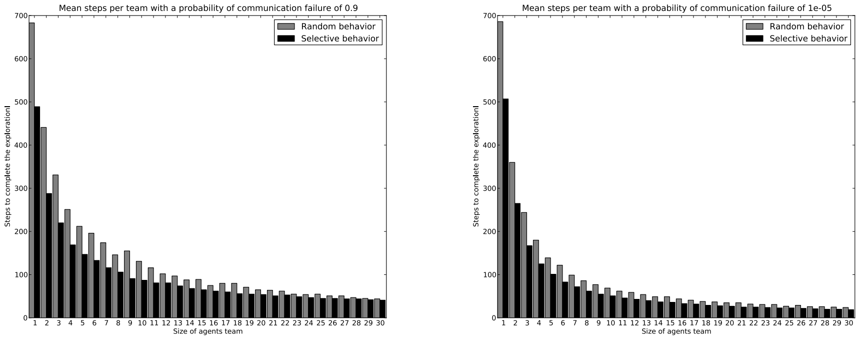


Fig. 6. Experiments results relative to mean steps

In this case, the improvement over the random behavior is not so good as in previous experiments. Now, as is showed in Fig. 5 (b), it ranges 40% to 20% (10% for a high number of agents per team and a high probability of communication failure).

If we compare our method in both maps (Fig. 7), it can be appreciate as the edge increase does not significantly affect the steps necessary for complete inspection of the environment, i.e. our method is quite robust against the map topology.

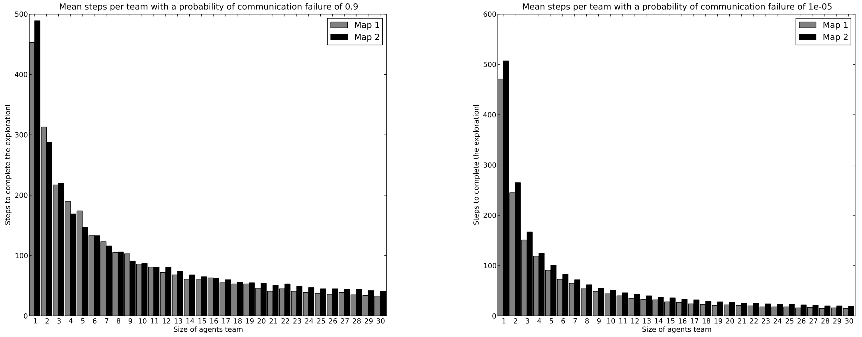
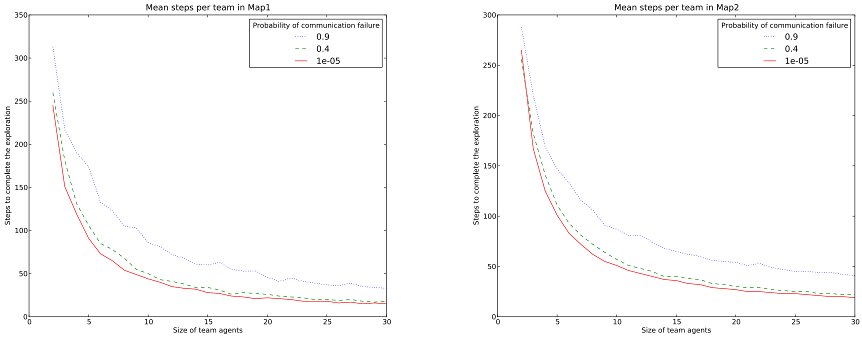


Fig. 7. Comparative between the two maps and the method presented

Finally, Fig. 8 displays how the probability of communication failure affects the results. It can be noticed that the average steps needed for a complete inspection process when the communication failure probability is intermediate are very close to the steps needed when the probability of communication failure is very small, i.e. our method is tolerant to communication failure.



(a) *Map 1*

(b) *Map 2*

Fig. 8. Comparative between the two maps and different probability of communication failure

4 Conclusions and Future Work

A novel, selective method based on auctions for map inspection by teams of robotic agents has been proposed. The experiments conducted show that the selective method improves a random behavior, which could be very useful in places with communication problems. The method is able to inspect in finite time, pipe or sewer environments modeled as a graph. A provided framework for the coordination of members of a team of robots. We have emphasized on to solve the communication problems, which are inherent to this kind of environments.

Currently, we are working on the improvement of route selection method, seeking for optimal routes that will be centered in complete an intelligent exploration of the immediate environment. Likewise, the inclusion of other aspects for the planned agent's route evaluation could be a good first step for future versions of this method. We are also addressing the exploration of unknown environments with this method.

References

1. Akyildiz, I., Sun, Z., Vuran, M.C.: Signal propagation techniques for wireless underground communication networks. Physical Communication 2(3), 167–183 (2009)
2. Breazeal, C.: Emotion and sociable humanoid robots. International Journal of Human-Computer Studies 59(1-2), 119–155 (2003)
3. Burdakov, O., Doherty, P., Holmberg, K., Kvarnström, J., Olsson, P.M.: Positioning unmanned aerial vehicles as communication relays for surveillance tasks. In: Proceedings of the 5th Robotics: Science and Systems Conference (RSS), Seattle, WA. Citeseer (2009)
4. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration. IEEE Transactions on Robotics 21(3), 376–386 (2005)
5. Chehri, A., Fortier, P., Tardif, P.: Application of Ad-hoc sensor networks for localization in underground mines. In: IEEE Annual Wireless and Microwave Technology Conference, WAMICON 2006, pp. 1–4. IEEE, Los Alamitos (2007)
6. Dasgupta, P.: A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 38(3), 549–563 (2008)
7. Duro, R.J., Graña, M., Lope, D.: On the potential contributions of hybrid intelligent approaches to multicomponent robotic system development. Information Sciences 180(14), 2635–2648 (2010)
8. Emery, R., Sikorski, K., Balch, T.: Protocols for collaboration, coordination and dynamic role assignment in a robot team. In: Proceedings of IEEE International Conference on Robotics and Automation ICRA 2002, vol. 3, pp. 3008–3015. IEEE, Los Alamitos (2002)
9. Kawaguchi, Y., Yoshida, I., Kurumatani, H., Kikuta, T., Yamada, Y.: Internal pipe inspection robot. In: Proceedings of IEEE International Conference on Robotics and Automation, 1995, vol. 1, pp. 857–862. IEEE, Los Alamitos (2002)
10. Kurnaz, S., Cetin, O., Kaynak, O.: Fuzzy logic based approach to design of flight control and navigation tasks for autonomous unmanned aerial vehicles. Journal of Intelligent and Robotic Systems 54(1), 229–244 (2009)
11. Maio, D., Rizzi, S.: A multi-agent approach to environment exploration. IEEE Transactions on Pattern Analysis Machine Intelligence 18(11) (1996)
12. Mathis, P.: Graphs and Networks: Multilevel Modeling (Geographical Information Systems Series) (2007)
13. Osuka, K., Kitajima, H.: Development of mobile inspection robot for rescue activities: MOIRA. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2003, vol. 4, pp. 3373–3377. IEEE, Los Alamitos (2003)

14. Robins, B., Dautenhahn, K., Boekhorst, R., Billard, A.: Robotic assistants in therapy and education of children with autism: Can a small humanoid robot help encourage social interaction skills? Universal Access in the Information Society 4(2), 105–120 (2005)
15. Rome, E., Hertzberg, J., Kirchner, F., Licht, U., Christaller, T.: Towards autonomous sewer robots: the MAKRO project. Urban Water 1(1), 57–70 (1999)
16. Shoham, Y., Leyton-Brown, K.: Multiagent systems: algorithmic, game-theoretic, and logical foundations. Cambridge University Press, Cambridge (2008)
17. Silva, A., Vuran, M.: Empirical evaluation of wireless underground-to-underground communication in wireless underground sensor networks. Distributed Computing in Sensor Systems, 231–244 (2009)
18. Thrun, S., Thayer, S., Whittaker, W., Baker, C., Burgard, W., Ferguson, D., Hahnel, D., Montemerlo, D., Morris, A., Omohundro, Z., et al.: Autonomous exploration and mapping of abandoned mines. Robotics & Automation Magazine 11(4), 79–91 (2005)
19. Zhang, Y., Yan, G.: In-pipe inspection robot with active pipe-diameter adaptability and automatic tractive force adjusting. Mechanism and Machine Theory 42(12), 1618–1631 (2007)

Study of a Multi-Robot Collaborative Task through Reinforcement Learning

Juan Pereda[1], Manuel Martín-Ortiz[1], Javier de Lope[2], and Félix de la Paz[3]

[1] ITRB Labs
Research, Technology Development and Innovation, S.L.
[2] Computational Cognitive Robotics
Universidad Politécnica de Madrid
[3] Dept. Artificial Intelligence
UNED
{juan.pereda,manuel.martin}@itrblabs.eu, javier.delope@upm.es,
delapaz@dia.uned.es

Abstract. A open issue in multi-robots systems is coordinating the collaboration between several agents to obtain a common goal. The most popular solutions use complex systems, several types of sensors and complicated controls systems. This paper describes a general approach for coordinating the movement of objects by using reinforcement learning. Thus, the method proposes a framework in which two robots are able to work together in order to achieve a common goal. We use simple robots without any kind of internal sensors and they only obtain information from a central camera. The main objective of this paper is to define and to verify a method based on reinforcement learning for multi-robot systems, which learn to coordinate their actions for achieving common goal.

1 Introduction

One of the major problems that we have to deal is to achieve that robots collaborate in order to obtain a common objective. Part of the approach present separate mechanisms for controlling the robot and for defining the coordination process. This kind of solutions has the problem that the robot control interferes in the solution adopted by the team.

Matarić [4] considers that some problems of coordination multi-robot systems is an interested area of research. She present some approach for multi-robot collaborative situations using reinforcement learning for collective foraging [5].

Some modern studies, use hybrid controllers based in evolutionary algorithms and reinforcement learning for autonomous robots [7]. This approach use the optimization force of the evolutionary algorithm and the efficiency of the reinforcement learning to resolve the navigation problem in environment with obstacles. This kind of solutions are designed for single robots but do not analyze the relation between different robots to do collaborative tasks.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 185–191, 2011.

Other studies like [1], present new approach for the reinforcement learning using k-NN method to improve converge and memory use, two of the critical points in this kind of algorithm.

Normally, the different studies present reinforcement learning only for individual learning. Some theories propose co-learning solutions for the multi-robots environments [3] based on the difficulties of scaling up the individual solutions to multi-robots one. Other approach with multi-robot systems [2] presents solutions with two learning strategies: one individual and another one for the collaborative tasks.

In this paper, we define a method for solving collaborative tasks by means of reinforcement learning. In the sequel we propose the basics and expose the experimental results.

2 Collaborative Object Movement

2.1 Description of the Experimental Task

The problem we want to resolve is to move a rectangular object from an initial random position to a goal position that we define at the beginning of the experiment.

We define a environment without obstacles, with two robots and one rectangular object that the robots have to move from initial point to the goal. We use two robots that do not have any kind of sensory input the system know the robot position due to we are using a cenital camera. With this camera, the system knows where the object in and the other members of the team are. At this stage, we do not employ the effects of the natural forces. We consider just four basic movements for the robots: advance, turn left, turn right and go back. They do not have any kind of arm or external actuator to move the objects, they only can push it.

Figure 1 shows a setup representation where the two robots (represented by two circles) move the object (represented like a stick) to a predefined goal (represented with a rectangles). The triangular object represents the camera that provides information of the environment to the robots.

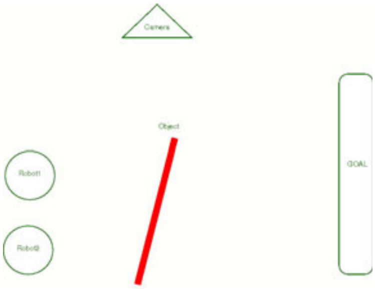


Fig. 1. Experiment setup

2.2 Q-Learning Method

Reinforcement learning uses a trial and error method to learn how the agents have to do the tasks. The agents interacts with the environment and receives rewards or punishments depending on the actions result [8,6].

We use the Q-learning algorithm to identify the actions that the robots must apply in each particular state in order to perform the task. We use the conventional Q-learning update function [6,8]:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\,[\,r + \gamma\, max_{\alpha} Q(s',a') - Q(s,a)] \tag{1}$$

2.3 States and Actions

To solve our problem we consider two different parts in the object: P and P'. These parts define where the robots must push to move it. Figure 2 and figure 3 depict representation the object the differents movements of the object when the robots push it. The grayed zone represents the object goal position. The circles represents the robots. The arrows represent the robot's push movement.

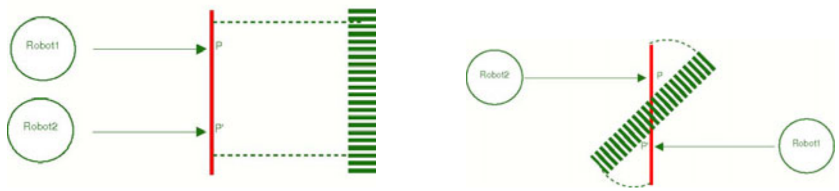


Fig. 2. Definition of states and actions. a) Decrease distance. b) P decrease, P' increase.

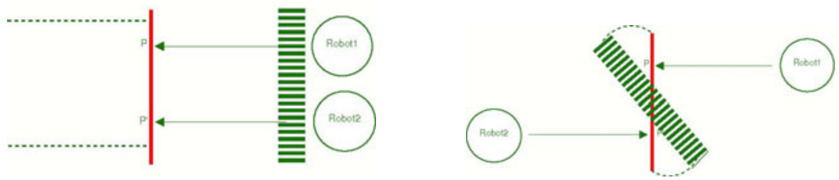


Fig. 3. Definition of states and actions. a) Increase distance. b) P increase, P' decrease.

There are four different object movements according to where the robots push it. If the two robots push to the same way, the object will advance or go back. If the robots push it in the opposite sides the object turn to the left or to the right over it self. So, if we consider these four kinds of movements we define four states in our problem:

1. The distances between both P and P' and the goal decrease. (Figure 2 a)
2. The distances between both P and P' and the goal increase. (Figure 3 a)

3. The distance between P and the goal decreases, and the distance between P' and the goal increases. (Figure 2 b)
4. The distance between P and the goal increases, and the distance between P' and the goal decreases. (Figure 3 b)

In the same way, we define four possibles actions to obtain these different states that stablish the collaboration between two robots (Robot 1 and Robot 2):

1. Robot 1 and Robot 2 push in the same direction. They push towards the goal.
2. Robot 1 push in the goal direction and Robot 2 push in the opposite direction.
3. Robot 1 push in the opposite direction and Robot 2 push in the goal direction.
4. Robot 1 and Robot 2 push in the same direction and they push in the opposite direction to the goal.

We consider that the robots do not push to the same part at the same moment, i.e, if the Robot 1 push in P part then the Robot 2 has to push in the P' part necessarily. As we said above, the two robots have the same configuration so it does not matter which robot pushes in which part.

3 Experimental Results

3.1 Nine States Configuration Approach

In these experiments we introduce some special states but conserves the same number of possible actions that we have defined in the 2.3 subsection. We are going to consider five new "no move" states. These new states are:

1. The distance between both P and P' and the goal is the same.
2. The distance between P and the goal is the same, and the distance betweenP' and the goal decrease.
3. The distance between P and the goal is the same, and the distance betweenP' and the goal increase.
4. The distance between P and the goal decrease, and the distance between P' and the goal is the same.
5. The distance between$P$ and the goal increase, and the distance between P' and the goal is the same.

We start our experiments taking the decision of evaluate every single step and if the distance to the goal decrease we reward the action. With this approach we define a matrix of rewards and update the Q matrix in all steps. With this approach we have good results in the experiments but only in the cases where the goal and the stick are in parallel path. Another problem with this approach is to adjust the rewards matrix. So based in Sutton and Barto studies [6,8], we decide to evaluate the complete episode and only award the behavior that finish with the stick in the goal. Now we only have two cases:

1. If the episode finish at the goal give a reward.
2. If the episode finish with the limit of the steps we penalize.

Later, when we have the reward for the episode, we go all over the steps and calculate, applying equation1, the Q value for the differents steps. So we reward all the steps that are involved in the correct episode.

In the figure 4we can see one example episode. In the figure, we show the last episode where the robots push the stick from the initial position to the goal represented with a green line.

In the figure 5 we can see the result for the experiment with nine states. The convergence take place before the 500 episode.

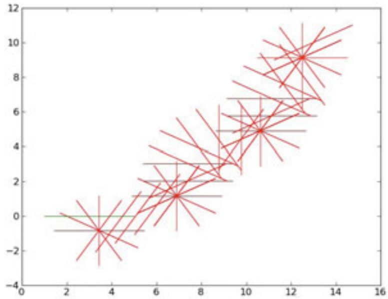


Fig. 4. Examples from one learning episode

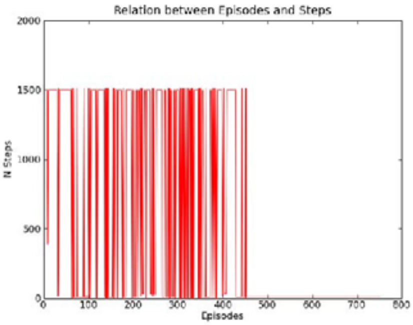


Fig. 5. Nine states convergence curve

3.2 Four States Refinement

We analyze the environment and observe that we can simplify the environment definition in four states. We define the relation between the different states as we can see in the figure 6. If we simplify in four states we can obtain the convergence in too many episodes without losing any quality in the solution.

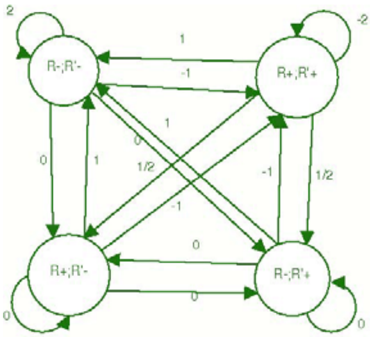


Fig. 6. Four state diagram

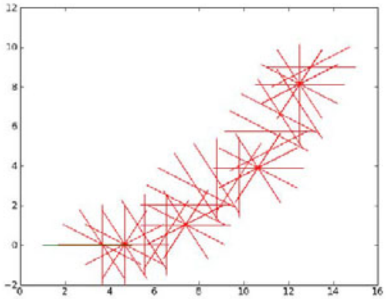


Fig. 7. Results four states

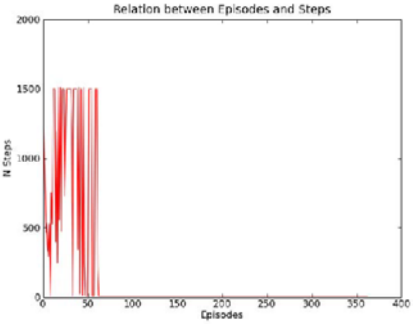


Fig. 8. Four states convergence curve

In the results we observe the differences between nine states and this other simple approximation and see that it does not affect to the quality of the experiments results. In the figure 7we show the last episode for the same experiment that we run with nine states approach. We can see that the result is similar than the result of the figure 4 so we can say that we do not lose any quality in the solution. However, the number of episodes that we need to converge is reduced to only 50 episodes as we can see in the figure 8.

4 Conclusion and Future Works

A simple solution to coordinate two robots for moving an object to a final goal position has been presented. The solution learns for different cases and it gives a more robust and simple solution. We have considered two solutions based on the reward at the end of the episode. The second approach considers a smaller set of states, which helps to produce an early converge. As we explain we do not need to use nine states —the first approach— to resolve our reinforcement learning problem. We only consider the nine states in case we use more possible actions that include stop one for the robots.

We are going to advance in two different lines. First, we are going to employ the Webots simulator [9] in order to get a more realistic results. Then, we will use real robots in a real environment. We are already developing some procedures for computing the robots and stick positions by using an external vision system. Also, we are going to explore other reward/punishment schemes. Particularly we want to study if a specific reward for the stick orientation could improve the general performance.

References

1. Martin, J.A., de Lope, J., Maravall, D.: Analysis and solution of a predator-protector-prey multi-robot system by a high-level reinforcement learning architecture and adaptive systems theory. Neurocomputing 58(12), 1266–1272 (2010)
2. Iima, H., Kuroe, Y.: Swarm Reinforcement Learning Algortithms Based on Sarsa Method. In: SICE Annual Conference (2008)
3. Yang, E., Gu, D.: Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey. CSM-404. Technical Reports of the Department of Computer Science, University of Essex (2004)
4. Matarić, M.J.: Coordination and learning in Multi-Robot Systems, pp. 6–8. IEEE Computer Society Press, Los Alamitos (1998)
5. Matarić, M.J.: Reinforcement Learning in the Multi-Robot Domain. Autonomous Robots 4(1), 73–83 (1997)
6. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
7. Maravall, D., De Lope, J., Martín H, J.A.: Hybridizing evolutionary computation and reinforcement learning for the design of almost universal controllers for autonomous robots. Neurocomputing 72(4-6), 887–894 (2009)
8. Sutton, R.S.: Reinforcement learning architectures. In: Proc. Int. Symp. on Neural Information Processing, Kyushu Inst., Kyushu Inst. of Technology, Japan (1992)
9. Webots. Commercial Mobile Robot Simulation Software, `http://www.cyberbotics.com`

Design of Social Agents

Roman Gorbunov, Emilia Barakova, and Matthias Rauterberg

Eindhoven University of Technology, Den Dolech 2, Postbus 513, 5600 MB
Eindhoven, The Netherlands

Abstract. Social behavior, as compared to the egoistic and rational behavior, is known to be more beneficial to groups of subjects and even to individual members of a group. For this reason, social norms naturally emerge as a product of evolution in human and animal populations. The benefit of the social behavior makes it also an interesting subject in the field of artificial agents. Social interactions implemented in computer agents can improve their personal and group performance. In this study we formulate design principles of social agents and use them to create social computer agents. To construct social agents we take two approaches. First, we construct social computer agents based on our understanding of social norms. Second, we use an evolutionary approach to create social agents. The social agents are shown to outperform agents that do not utilize social behavior.

1 Introduction

Collaboration, trust and other social norms are crucial for good operation of teams. Our long term goal is to create a model of social norms that will enable prediction of behavior and interactions within a team. On the basis of such a model we want to test hypotheses of how these interactions within a group will develop.

Human and animal models of behavior inspired studies on social learning [4,19,18,17,21] and models of social norms such as trust and collaboration [19,18,17,14]. Collaboration has also been studied within economics, sociology, and game theory. In economics, models of a rational actor who collaborates to improve organizational effectiveness have been developed [15,11,9]. In sociology [5], the non-rational and social aspects of collaboration has been explored. Gorbunov at al. [8] have proposed a model that accounts for the deviations from the rational behavior.

Games, in particular are an establishing field that makes possible realistic social interactions to be exposed, monitored and trained [2,1,3,12,13,16]. Hennes [12] Voynarovskaya et al.[20] and Gorbunov at al. [8] have developed a game-based method for monitoring the social relations in small groups of individuals. It is based on the records of a negotiation game that monitors of the collaborative behavior of the players. The Colored Trails game framework [7] in a three-player negotiation variation [6,20] was used for this study. The conclusions of the study have been restricted by the limited amount of data and the

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 192–201, 2011.

difficulties to constrain the experiment by having a control over one or several individuals.

We propose to use a multi-agent system approach for the purpose of creating an agent-based support tool for team collaboration as a first step towards prediction of conflicts in small teams of people. We design the agent-players that in controlled manner can express collaborative or egoistic behavior. We hypothesize that the agent with persistent collaborative behavior will be more successful in collaboration game. We also expect that the outcome of the experiments will give us better insights of how to improve the agents so they can be used as a controlled substitute of a human player.

Moreover, we want to understand the mechanism of collaborative behavior, i.e. how collaboration is maintained, how to avoid exploitation of collaborative agents. This understanding will help us interpret the data from MARS 500 experiment [12,20,10].

2 Methods

2.1 Settings of the Interactions

In our study we use the prisoner dilemma, a classical problem in the game theory, which used to study collaborative behavior. Every player can chose either cooperation or defection. If both players choose to cooperate, than they get 3 points each. In contrast, every player gets only 1 point if both of them choose to defect. In other words, the mutual cooperation is more beneficial than the mutual defection. On the other hand, the defection is always more beneficial than then cooperation under any given strategy of the partner. If one player defects and another one cooperates than the defecting and cooperating player get 5 and 0 points, respectively. Because of the above mentioned properties of the game, players face a dilemma: to cooperate or to defect.

In our study the interaction between two players (computer agents) is organized in the following way. One of the agents starts the game by choosing one of the two available moves (cooperation or defection). Another agent "sees" the move chosen by the first agent and uses this information to make its own move. After two agents made their moves, each of them gets points determined by the above given payoff matrix. The third move is made by the first agent. The third and the second moves together determine another portion of points that will be delivered to the two interacting agents. In other words, the second move determines the payoffs two times (first, in combination with the first move and then in combination with the third move). The agents make subsequent moves in the above described way until a predefined number of moves is made.

2.2 Model of the Agents

To formalize algorithms imbedded into the computer agents we consider a model similar to the Turing machine. Every agent has a fixed and limited number of internal states. The previous move of the partner is considered as the input for the algorithm. The input and the internal state of the agent determine its output (the next move) that can be either cooperation or defection. The mapping of the inputs and internal states into the output is given by a fixed matrix. Additionally to the generation of the output the agent changes its internal states. The new internal state, like the output, is determined by the input and previous internal state. The mapping of the input and the current internal states into the new internal states is also given by a fixed matrix.

2.3 Design Principles

The maximization of the personal benefit of an agent in a sequence of interactions with a random partner is the main design criterion that we used for the construction of the social computer agents. In other words, our intention is to create an egoistic agent whose the only goal is to maximize its own benefit in the environment populated by other agents whose playing strategies are not known.

In the prisoners dilemma the defection is always the most beneficial choice independently on what choice, defection or cooperation, is made by the partner. As a consequence, the agents, that are rational in the classical game theory sense, always choose to defect. In other words, the rational agents demonstrate mutual defection that is known as Nash equilibrium for the prisoners dilemma. Further on we will call the agents of this kind as "defective" agent.

The permanent defection is one of the two simplest possible strategies and the permanent cooperation is the second one. We will call the agents that permanently cooperate as "cooperative" agents. This strategy can also be considered as a rational one since the mutual cooperation is more beneficial than the mutual defection.

However, it has to be emphasized that the cooperation is beneficial only if it is reciprocal. The cooperation with an agent that defects can be considered as an exploitation of the cooperative agent by the defective one. This kind of interaction is very beneficial to the exploiting agent and very unbeneficial to the exploited one. Thus the cooperative agent cannot be considered as a successful solution for the environment populated by agents of different kinds.

To prevent an exploitation by a partner the agent should exhibit a reciprocal behavior (defect with the defective agents and cooperate with the cooperative agent). We will call the agents of this type as "reciprocal" agents. The reciprocal agents can be considered as more successful than the above introduced defective and cooperative agents. In contrast to the cooperative agents, the reciprocal agents cannot be exploited by the defective agents since they will behave defectively with the defective agents. In contrast to the defective agents, the reciprocal agents will establish and maintain cooperation with each other if they are constructed to start from the cooperative move. The weak point of the reciprocal agents is that they are not able to exploit cooperative agents. In other words, the reciprocal agent cooperates with its partner even if it is not a necessary condition for the cooperation of the partner. This kind of behavior is in a disagreement with the above introduced goal to find a behavior that maximizes the personal benefit of the agent.

The above described requirements can be summarized in the following list of the social skills that have to be embedded into the social agents:

- To avoid the exploitation by the partner, the social agent should be able to reciprocate defection of the partner (react defectively on the defection).
- To maintain the cooperation, the social agents should be able to reciprocate the cooperation of the partner (react cooperatively on the cooperation).
- To establish the cooperation, the social agents should be able to propose the cooperation in the phase of the mutual defection (react cooperatively on the defection).
- To exploit the partner, the social agents should be able to try defection in the phase of the mutual cooperation (react defectively on the cooperation).

As we can see, the above described requirements can be competing. So, the agent needs to be able to decide what logic to apply at the current stage of the interaction. Moreover, the agent needs to remember the responses of the partner. In more details, it should not keep trying to establish a cooperation if all these attempts fail. Similarly, the agent should not continuously try to exploit its partner if it sees that these attempts are not successful. On the other hand, the agent should not cooperate if it is not required for the cooperation of the partner.

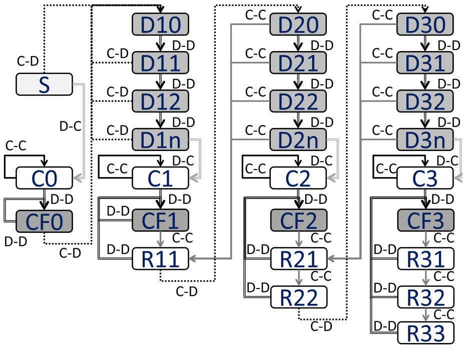


Fig. 1. Schema of the constructed social agent

We implement the above described design principles by the following algorithm. The agents start cooperation by proposing cooperative move as a response on the defective move of the partner. Doing that agent switches to the state of cooperation (CN in the Fig. 1). If partner answers defectively on cooperation initiated by the agent, the agents switches to the cooperation-failed (CF_n) states. In this state the agents never initiates a new cooperation. If partner starts cooperation, the agent will defect, immediately or after several steps of cooperation, to check if its partner can be exploited. In more details, for several steps agent plays only defectively. If at this stage the partner made at least one cooperative move, the agents concludes that exploitation was successful and try to exploit the partner again. If the agent has already tried to exploit after n steps of cooperation and this attempt failed, the next attempt to exploit will start after $n+1$ steps of cooperation. If the agent tried to exploit after N_{max} steps of mutual cooperation and this attempt failed, it never tries to exploit again. After every unsuccessful attempt of exploitation agents tries to establish cooperation. More details about the algorithm are given in the Tab.1. In this table different columns and rows represent different inputs (moves of the partner) and internal states of the agents, respectively. The cells of the tables contain outputs (moves of the agents) and a new state of the agent. The defective and cooperative moves are denoted as $\ominus$ and $\oplus$, respectively.

Table 1. Algorithm of the Constructed Social Agent

Internal State	Cooperative input	Defective input	No input
S	$\ominus$, D10	$\oplus$, C0	$\oplus$, C0
D10	$\ominus$, D10	$\ominus$, D11	
C0	$\oplus$, C0	$\ominus$, CF0	
D11	$\ominus$, D10	$\oplus$, C1	
CF0	$\ominus$, D10	$\ominus$, CF0	
C1	$\oplus$, C1	$\ominus$, CF1	
CF1	$\oplus$, R11	$\ominus$, CF1	
R11	$\ominus$, D20	$\ominus$, CF1	
D20	$\oplus$, R11	$\ominus$, D21	
D21	$\ominus$, D20	$\oplus$, C2	
C2	$\oplus$, C2	$\ominus$, CF2	
CF2	$\oplus$, CF2	$\ominus$, CF2	

2.4 Evolution of Agents

In our study we apply an evolutionary optimization to find a well performing agent that can be compared with the constructed one. The evolution starts from a random agent containing the same number of states (12) as the constructed one. At every step of the evolution we perform a random mutation of the agent by changing one element in one of the two matrices that define the agent behavior. If the mutated agent performs better as the parent agent, we replace the parent agent by its child and continue the procedure.

3 Results

An interaction between any two agents can be described by the ordered sequence of moved made by the two agents. If the interaction between the two fixed agents is long enough, the sequence of moves converge to a periodic pattern (because of the limited number of the internal states of the agents). As a consequence the performance of the agents, which is defined as the average score per game, depends on the length of the interaction. In particular it is meter if the convergence is reached and, if is the case, at which phased of the periodic patter the interaction is stopped. To get a representative performance of an agent we have to make sure that the interaction between the agents is long enough to suppress the dependency of the performance on the length of the interaction.

The performance of two agents can also depend on which agent started the interaction first. To deal with this ambiguity, for any pair of agents we run two sequences of interaction such that every agent starts one sequence.

For 29 agents we run the interactions containing 1000 steps. We found out that for all considered agents the difference between the performances on the step 999 and 1000 do not exceed $1.58\ 10^{-3}$. This number is small in comparison with the variation of the performances of different agents, indicating that 1000 steps are sufficient to reach the convergence. Further on we will use this number of games to calculate performance of agents, unless otherwise stated.

The performance of an agent also depends on the partner. Moreover, even if agent A outperforms agent B while both playing with partner C it is still possibe and agent A underperforms B while both playing with another partner D. To get a representative performance of an agent we create a larger pool of random partners and let the agent to play with every partner from the pool. Then the performances of the agent with different partners are averaged to get the representative performance. In our study we populated the pool by agents that have no more than 12 internal states. The probability of an agent to be in the pool does not depend on the number of internal states.

We chose the size of the pool of partners in the following way. For a given random agent we have generated 10 different pools of the same size. Every pool has been used to calculate the performance of the given agent and then the difference between the maximal and minimal performance was calculated. If this difference was larger than 0.1 we double the size of the pool and repeat the same calculations. We start the procedure from the minimal possible pools containing only one partner and grow the pool in the above described way until the difference between the minimal and maximal performances of the considered agent is smaller than 0.1. The above procedure has been performed for 7 different random agents. We found out that for the all considered agents the range of the performances became smaller than 0.1 with the pool containing less than 10 thousands agents. Further on pools containing 10 thousands agents are used, unless otherwise stated.

We used the above given values for the length of the interaction and the size of the pool to estimate the performance of the constructed social agent. We found out that the score per game for this agent is equal to approximately 3.16. To get an idea of how good this performance is we have calculated performances of 4855 randomly generated agents. The distribution of the performances of these agents is shown in the Fig.2. As we can see in the figure, the constructed social agent outperform majority of randomly generated agents. In more details, approximately 97.8 % of random agents were outperformed by the constructed agent.

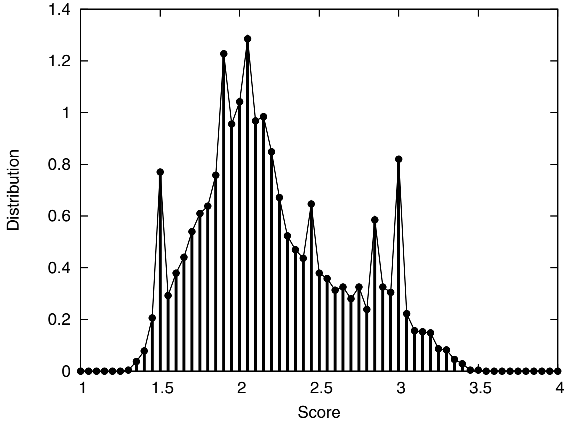


Fig. 2. Distribution of the performances of the random agents

However, we have to note that the performances calculated in this way should depend on the way the pool is populated. In our study we used a procedure in which any two agents, that have the same number of the internal states, have the same chance to be put into the pool. Moreover, the probability of an agent to be added to the pool does not depend on the number of the internal states. In other words we got a "uniform" distribution of the agents in the pool and, as a consequence, the performance of agents, calculated in this way, can be considered as more "representative" or "universal".

However, we need to remember that the assumption of the homogeneous distribution of the agents in the pool is not always valid. In particular, the pool can be populated by artificial agents that were constructed with the intention to increase the personal performance. Or, as another example, the content of the pool can be a product of an evolution. In both cases the chances of an agent to be in the pool are higher if its performance is better.

To overcome the problem of the dependency of the score on the way the pool of partners was generated, we propose to compare performances of two agents by letting them to interact with each other. In more details, we want to model performances of two different agents in the pool that is equally populated by the agents of the two considered types. To model this situation we do not need to use a larger pool because the interactions of a given agent with the partners of the same type will always give the same score per game. The agent of a chosen type needs to interact with one partner of the same type and one partner of another type. Interactions with partner of two different types of agents gives two values of the performance that are then averaged to get the performance of the given agent in the considered situation. The performances of the two agents, calculated in this way, can be interpreted as their abilities to survive, in the evolutional sense, in the pool that is equally populated by the agents of the two considered types. We would like to emphasize that the performances of agents, calculated in the above described way, are not transitive. If agent A outperforms agent B and agent B, in its turn, outperforms agent C, it does not necessarily means that agent A outperforms agent C. As the consequence, the second way to compare performances of agents cannot be used in the evolutionary optimization.

We used the above described procedure to estimate performances of the constructed and evolved agents. The constructed social agent has been compared with 10 thousands randomly generated agents. In more details, for every pair of the random and constructed agent we calculated the performances of the both agents while interacting with each other. The obtained performances are shown in the Fig.3. The x-axis and y-axis correspond to the scores of the constructed and random agents, respectively. As we can see in the figure the performance of the constructed agent was always better than those of the confronted random partner. This represents the fact that the constructed agents cannot be exploited by their partners.

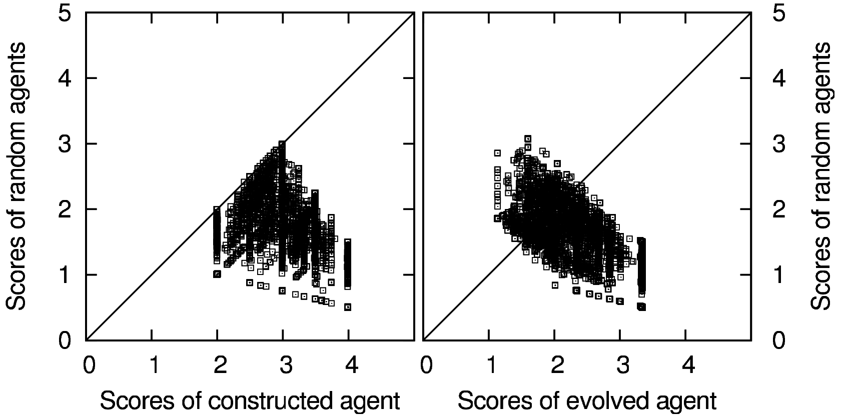


Fig. 3. Performances of the constructed and evolved agents compared with the perfomrances of the random agents

In the same way we have estimated the performance of the best agents found by the evolutionary optimization (see Fig.3). As we can see in the figure, the random agents outperformed the evolved one in many cases. In other words the evolved agent does not have mechanisms that protect it from the exploitation. As we have shown before, the evolved agent significantly outperformed the constructed one in the pool of the random partners (with the average score per game equal to about 3.6 and 3.2 for the evolved and the constructed agents, respectively). However, in spite of that fact, we found out that in the pair interactions with the random agents the average performance of the evolved agent was smaller than those of the constructed agent (about 2.6 and 3.1, respectively). It can be explained by the fact that the constructed agent interact with each other better than the evolved agents. The average performance of the constructed agents in the pool of the constructed partner is about 3.0 while the evolved agents in the pool of the evolved partner have performance equal to about 1.7.

4 Discussion

In our study we provided computer agents that are able to increase personal performance by utilizing social interactions with random agents with unknown strategies. In more details, we provided a mechanism that allows agents to establish and maintain cooperation. The presented agents establish and maintain cooperation only if it is necessary and sufficient condition for the cooperative behavior of the partner. Moreover, the constructed agents are able to prevent exploitation by the defective partner. And finally, the presented agents can exploit those partners that allow the exploitation. We have demonstrated that the constructed agents perform outperform majority of random agents and even the agents that have been obtained by evolutionary optimization. In this way we proposed the computational model that describes cooperation and exploitation.

The proposed model provides a mechanism of how collaboration and exploitation can be established and maintained. As a consequence we contribute to a better understanding of these social phenomena. The presented model can be used to identify collaborative and exploiting patterns in humans behavior and, in this way, explain human behavior better. Moreover, the constructed agents mimic human social behavior and, as a consequence, can better play against human players in collaborative computer games.

Acknowledgments

The research reported in this paper is supported by NWO User Support Program Space Research. The project number is ALW-GO-MG/07-13.

References

1. Barakova, E.I., Gillessen, J., Feijs, L.: Social training of autistic children with interactive intelligent agents. Journal of Integrative Neuroscience 8 (1), 23–34 (2009)
2. Barakova, E.I., Lourens, T.: Expressing and interpreting emotional movements in social games with robots. Personal and Ubiquitous Computing 14, 457–467 (2010)
3. Bekker, T., Sturm, J., Barakova, E.: Design for social interaction through physical play in diverse contexts of use. Personal and Ubiquitous Computing 14 (5), 381–383 (2010)
4. Bonnie, K.E., Horner, V., Whiten, A., de Waal, F.B.M.: Spread of arbitrary conventions among chimpanzees: a controlled experiment. Proceedings in Biological Science 274(1608), 367–372 (2007)
5. DiMaggio, P.J., Powell, W.W.: The iron cage revisited: Institutional isomorphism and collective rationality in organizational fields. American Sociological Review 48, 147–160 (1983)
6. Ficici, S.G., Pfeffer, A.: Modeling how humans reason about others with partial information. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 315–322 (2008)
7. Gal, Y., Grosz, B.J., Kraus, S., Pfeffer, A., Shieber, S.: Colored trails: a formalism for investigating decision-making in strategic environments. In: Proceedings of the 2005 IJCAI workshop on reasoning, representation, and learning in computer games, pp. 25–30 (2005)

8. Gorbunov, R., Barakova, E., Ahn, R., Rauterberg, G.W.M.: Monitoring interpersonal relations through collaborative computer games (2011) (submitted)
9. Grant, R.M.: The resource-based theory of competitive advantage: Implications for strategy formulation. California Management Review 34, 119–135 (1991)
10. Gushin, V.: Psychological countermeasures during space missions: russian experience. Journal of Gravitational Physiology 9 (1), 311–312 (2002)
11. Hamel, G., Doz, Y.L., Prahalad, C.: Collaborate with your competitors, and win. In: Hamel, G., Doz, Y.L., Prahalad, C. (eds.) Harvard Business Review, pp. 133–139 (1989)
12. Hennes, D., Tuyls, K.P., Neerincx, M.A., Rauterberg, G.W.M.: Micro-scale social network analysis for ultra-long space flights. In: The IJCAI-09 Workshop on Artificial Intelligence in Space, Pasadena, California, USA (2009)
13. Hennes, D., Tuyls, K.P., Rauterberg, G.W.M.: State-coupled replicator dynamics. In: 8th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2009), pp. 789–796 (2009)
14. Lee, U., Magistretti, E., Gerla, M., Bellavista, P., Li, P., Lee, K.W.: Bio-inspired multi-agent collaboration for urban monitoring applications. In: Liò, P., Yoneki, E., Crowcroft, J., Verma, D.C. (eds.) BIOWIRE 2007. LNCS, vol. 5151, pp. 204–216. Springer, Heidelberg (2008)
15. Pfeffer, J., Salanick, G.R.: The external control of organizations. Harper and Row, New York (1978)
16. Rauterberg, M., Neerincx, M., Tuyls, K., van Loon, J.: Entertainment computing in the orbit. International federation for information processing 279, 59–70 (2008)
17. Tomasello, M.: The Cultural Origins of Human Cognition. Harvard University Press, Harvard (1999)
18. Vanderelst, D., Ahn, R.M., Barakova, E.I.: Simulated trust: Towards robust social learning. In: Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, pp. 632–639 (2008)
19. Vanderelst, D., Ahn, R.M., Barakova, E.I.: Simulated trust: A cheap social learning strategy. Theoretical Population Biology 76, 189–196 (2009)
20. Voynarovskaya, N., Gorbunov, R., Barakova, E., Ahn, R., Rauterberg, M.: Nonverbal behavior observation: Collaborative gaming method for prediction of conflicts during long-term missions. In: Yang, H.S., Malaka, R., Hoshino, J., Han, J.H. (eds.) ICEC 2010. LNCS, vol. 6243, pp. 103–114. Springer, Heidelberg (2010)
21. Whiten, A., Spiteri, A., Horner, V., Bonnie, K.E., Lambeth, S.P., Schapiro, S.J., de Waal, F.B.: Transmission of multiple traditions within and between chimpanzee groups. Current Biology 17 (12), 1038–1043 (2007)

Event-Based System for Generation of Traffic Services in Road Congestions

C. Sotomayor-Martínez[1] and R. Toledo-Moreo[2]

[1] Univ. Murcia, DIIC, Faculty of Computer Science, 30100 Murcia, Spain
[2] Technical Univ. of Cartagena, DETCP, Edif. Antigones, 30202 Cartagena, Spain
csotomayor@um.es,
rafael.toledo@upct.es

Abstract. When traffic grows and a large number of vehicles meet on the same road stretch, any unusual behavior of a vehicle causes a change on the scene, pushing the rest of vehicles to adapt to the new situation. In these cases, the challenge is not only to predict conflictive situations, but also to do it in time. This paper focuses on the detection of traffic congestions by means of analyzing the behavior of the vehicles that conform a scene as independent events that influence one another. A Complex Event Processing (CEP) engine is employed for basic event processing and the establishment of relations among events, raising up the awareness of the situation as the linked events induce more and more complex ones. Once the status of the carriageway is known, the traffic center can determine what to do to reduce congestion and launches services such as drive speed commands or alternative routes.

1 Introduction

Large amounts of messages are sent continuously through vehicular networks. The information present in these messages must be propagated to the corresponding processing centers in order to analyze the situation, make decisions and launch the necessary actions. Each message generated by a vehicle describing its state and behavior represents an event used as a system input. Some other events coming from different sources, such as traffic cameras or meteorological conditions, can be added to these vehicular events to asses their influence on the traffic. Taking into account the diversity of the sources and contents, an efficient mean to process all this efficiently becomes necessary.

In the last years, many investigations and management solutions are based on two concepts: Event Stream Processing (ESP) and Complex Event Processing (CEP). Both approaches are oriented to the generation of real time information, monitoring and management in environments with large numbers of events. While ESP focuses on working with a single flow of simple events, CEP makes it possible to work with different flows, usually called a cloud, that can be highly complex and that can allow the execution of mathematical algorithms [1].

The use of CEP and ESP in Intelligent Transportation Systems (ITS) architectures has been shown in recent works such as [2,3]. Both works suggest

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 202–211, 2011.

event-based architectures as the basis for traffic management systems. Those architectures collect data from multiple sources along the road, building up events of higher level of abstraction in order to decide which possible actions can be applied into a traffic problem.

In this paper a combination of CEP and ESP is proposed for the problem of processing and classifying events generated by diverse sources, detecting complex patterns and creating automatic outputs that raise the situation awareness. This awareness makes it feasible for a traffic control center to warn drivers by means of panels or directly through vehicular communications links based on cellular networks or wireless local area networks.

The system under consideration focuses on traffic congestion detection and launches different actions depending on the severity of the congestion. In addition to the informative messages, vehicles may receive some other services when subscribed. These messages can be formatted in text, graphic or voice mode. Geospatial services are web-based and compliant with the Open Geospatial Consortium (OGC) standards [4].

The rest of the paper goes as follows: Section 2 describes the sensors in the vehicle, its data fusion model and the set of kinematic vehicles models that represents the motion behavior of the vehicle. Sections 3 and 4 introduce the architectures for event modelling and processing. Section 5 presents services associated to traffic congestion. The implementation of the system in depicted in Section 6, and Section 7 concludes the paper.

2 Road Congestions

First step in order to detect congestion is to define what is a congestion. In the system under consideration it is assumed that there is a congestion when vehicles must drive at a speed lower than the minimum speed limit. In Spain this value is 60 km/h in highways. With this number, and the relation between feasible average speed and traffic density shown in Fig. 1, several congestion levels are proposed. The system will detect the traffic densities associated to these situations. For the case of roads with speed limit 120 km/h (solid line in Fig. 1), congestion levels are presented in Table 1.

Table 1. Congestion Levels

Density (vehicule/km/lane)	Congestion Level
20 — 30	Slow Traffic
30 — 40	Slight Congestion
40 — 50	Middle Congestion
50 — 60	High Congestion
> 60	Serious Congestion

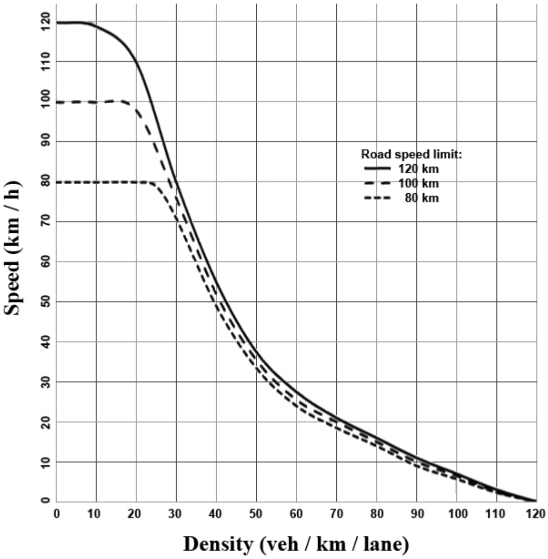


Fig. 1. Feasible average speed as a function of the traffic density

3 Event Modelling and Processing

The system receives as input a series of simple events generated by the vehicles. These events include the vehicle identifier, its position, velocity and maneuver, as described in previous Section.

These simple events, along with those coming from the infrastructure such as traffic cameras, meteorological stations, etc. are sent to the nearest base station that will perform a primary processing and send its gathered outputs to a traffic center, where complete processing is realized thanks to the information collected from the associated different geographical areas. At this level, it is possible to make decisions regarding traffic redistribution in a mandatory or voluntary manner, or send traffic patrols.

Fig. 2 shows the event flow among these entities and the actions which close the loop.

3.1 Event Processing

CEP technology can be applied to several ITS related fields. In this work it is exploited to raise the situation awareness in vehicle networks, processing events in base stations and traffic centers, and entering the feedback into the system.

A situation can be defined as a state of a set of entities at some particular time. It could be a vector of values of attributes belonging to one entity of multiple entities. It can be also described by means of the relations between one or multiple entities at some particular time. A simple event could be described by one entity, where one attribute, for example its velocity, changes forming a series

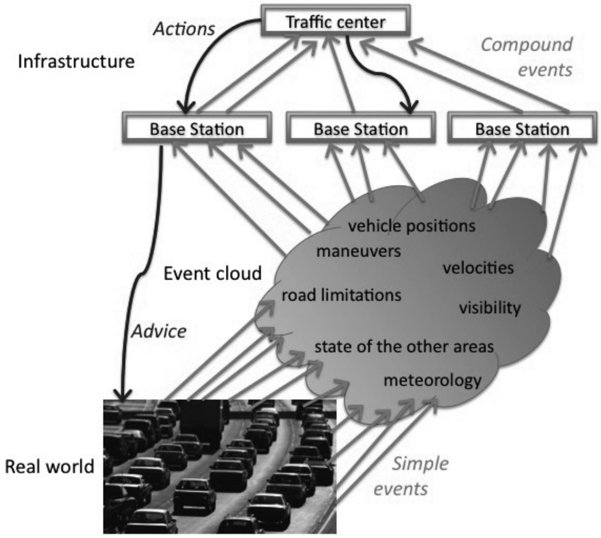


Fig. 2. Event flow among main entities of the system

of situations at each consequent time. More complex events are formed from the collection of more entities and their relations. Complex events are built from simpler events, applying analyzes of complex pattern detection, correlations, hierarchical rules and temporal relations of cause and belonging.

For the implementation of CEP modules in traffic centers, it was decided to employ Esper [5], a CEP/ESP engine in open code, implemented in Java and currently available over GPL license. Esper brings its own language for EPL queries (Event Processing query Language) based on SQL. Apart from launching continuous queries about event flows, this language makes it possible to launch triggers if a pattern has been detected in the event flow.

4 Traffic Congestion Detection Services

Open Geospatial Consortium (OGC) defines a series of standards for web-based services exploiting geographical information based on non-proprietary Internet standards, such as World Wide Web (WWW) of HTTP, Uniform Resource Locators (URLs), Multipurpose Internet Mail Extensions (MIME), Extensible Markup Language (XML) and other standards related to web-based services such as WSDL (Web Services Description Language) or SOAP (originally standing for Simple Object Access Protocol). These standards include Web Map Service (WMS) and Web Feature Service (WFS), employed in the implementation of the traffic center system under consideration.

WMS service produces dynamically a map with spatially referred data. This standard interprets a map as a representation of geographical information over a digital image file, and not the data themselves. Following this, traffic center provides thematic maps that include information regarding road occupation and can show alternative routes when destination has been settled.

On the other hand, the WFS service recovers geo-spatial data by means of independent calls to the platform. In the traffic center, it is used as a WMS complement, since makes it possible to check feature values, instead of show them graphically. For instance, with WMS can plot a map with the roads colored depending on their occupation, and WFS can report the number of vehicles per kilometer and the average speed or the length of a traffic jam in km.

5 System Architecture / Event-Based System

The system implementation follows the scheme shown in Fig. 3. On it, it can be observed that data coming from vehicle events or infrastructure are processed in the traffic center. Base stations are deployed along the road and work as intermediary between vehicles and traffic center in order to ease off the communications. They receive information coming from the vehicles in their corresponding environments and work as intermediary between vehicles and the traffic center. To do this, wireless-capable laptops were employed. Traffic center is in charge of most of the processing as it is detailed later.

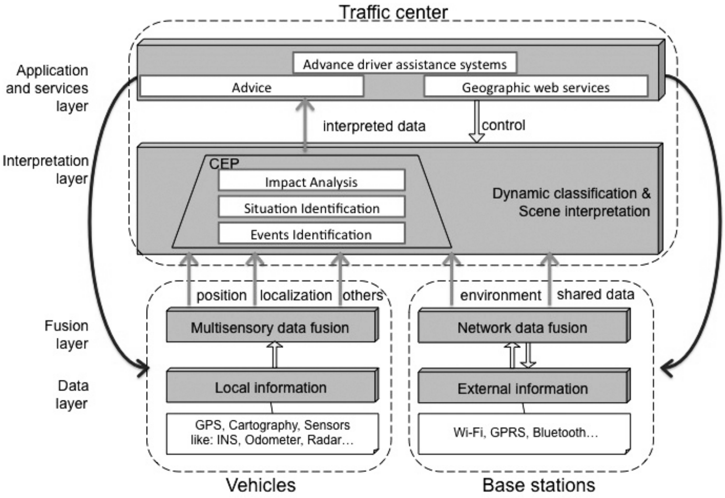


Fig. 3. Scheme of the system architecture

A prototype vehicle containing all necessary sensors and communication links was employed for the tests. For navigation purposes GPS, INS sensors and GIS maps are mounted board the vehicle. The use of inertial sensors, accelerometers and gyroscopes, supplies continuous positioning even in cases without GPS coverage, and fast dynamic measurements [7]. To integrate the data coming from

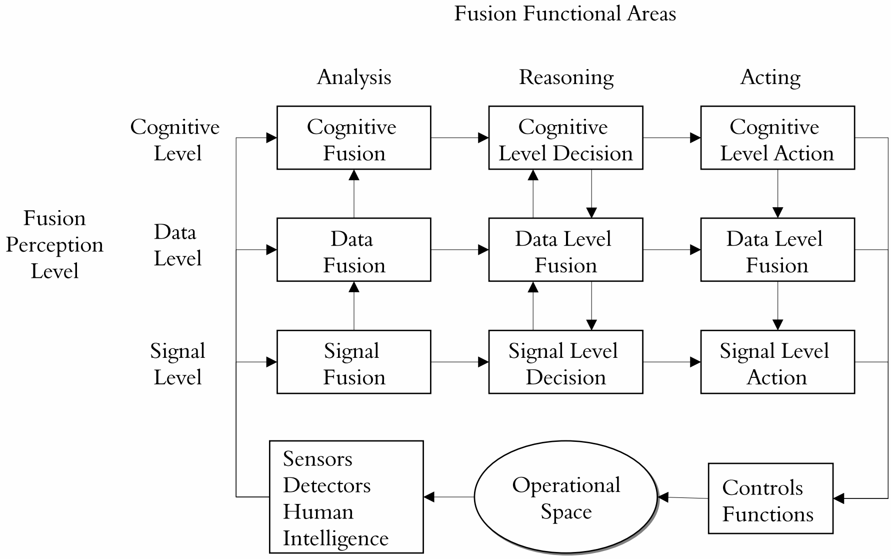


Fig. 4. Generalized fusion model

different sensors extended Kalman filters (EKF) run a set of different kinematical models as explained in Section 5.2. Thanks to the MEM (Micro-Electro-Mechanical) technology, low cost inertial sensors can be considered, at the expense of measurement noises and low level of performance [8]. To communicate the vehicles in the scene, WLAN ad hoc networks are used [9]. The test vehicle equipment is a Wireless LAN IEEE 802.11b and antenna by Cisco, EGNOS capable GPS and DGPS sensors by Novatel and Trimble, and MEM based IMUs by Crossbow and Xsens.

5.1 Fusion Model

Multiple sources fusion is approached in the current literature from many different points of view. Traditional fusion techniques dedicated to sensor measurements at data level of abstraction serve very usefully in many applications. Regarding road vehicles, many advanced driver assistance systems (ADAS) are possible thanks to multi-sensor data fusion filters. However, many applications require higher level of abstraction to describe situations in which a vehicle is involved. For military purposes, the issue of vehicle situation awareness has been approached by several authors from the point of view of the artificial intelligence [10]–[14]. Most of these authors agree to divide multi-sensor data fusion into four levels of increasing situation complexity. Some other approaches, like the one proposed by University of Melbourne, prefer different architecture schemes not necessarily oriented to military scenes [15]. Other authors such as [16] proposed a fusion architecture based on three levels of perception and functional areas, like the one shown in Fig. 4. In this model, vertical axis represents perception

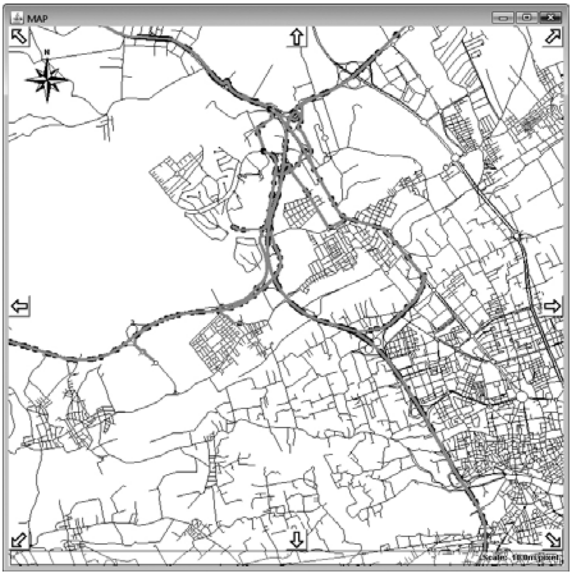


Fig. 5. Routes used for the system test

levels: signal, data and cognitive, while horizontal axis distinguishes among fusion functional areas: analysis, reasoning and acting.

5.2 Multiple-Sensor Data Fusion

Recently, many researchers have approached the inter-vehicles collision problem by using traditional data fusion techniques, and a number of different filters are used in the literature [17], [18]. Although, different techniques are being studied and analyzed by our group, to find a suitable estimate of the vehicle maneuver anytime, we will assume an IMM based method. The benefits of the interactive multiple model (IMM) based filter have been proved to be useful for navigation purposes [6]. In our case, the features of IMM based method are used for analysis and reasoning functional areas of the fusion model. The use of maneuver oriented models allow the determination of the vehicles roles in a situation. Basics of the IMM approach can be found in [19].

6 Implementation and Evaluation

The system has been implemented by means of three tools: Java Integrated Development Environment, Esper for creating and runing the CEP engine and SUMO for the generation of traffic simulations [20].

First tests were performed based on routes with the equipped vehicle prototype. Nevertheless, large scale tests are only affordable by means of traffic simulators, such as SUMO. Simulations in this work represent 30.000 vehicles driving more than 15 main routes and a number of alternative routes for each. Routes are depicted in Fig. 5. On this image, it can be seen how routes converge

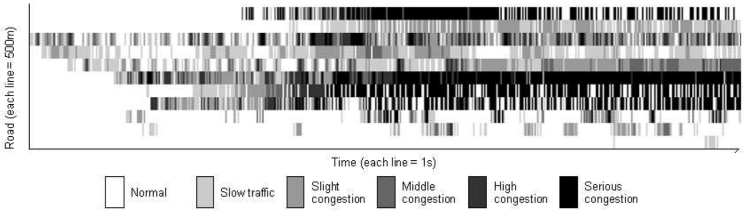


Fig. 6. Traffic density along time in test 1

between them, injecting traffic into some common stretches that will be the focus of our tests.

Among the parameters to be settled, let us remark the congestion level necessary for setting warning levels according to the severity of the situation. It has been also considered that not all the drivers are subscribed to the warning system, and not all of those who are subscribed will follow the system recommendations. This factor was introduced in the simulation by adjusting the percentage of changes in the routes and driving mode behaviors. Its influence must be analyzed in order to understand how this may affect the success of the system deployment.

For the same scenario, traffic simulations have been executed with the proposed system off and on (with different configurations and warning sets). Fig. 6 shows traffic density along a road time, presenting the traffic density in stretches of 500 m with different gray levels. Road direction is bottom-up in the image. Therefore, the motion of a vehicle on the road could be understood as a diagonal movement along the graph, from the bottom-left corner till the upper-right one.

In Fig. 6 for test 1 it can be seen how congestion grows in both severity, taking into account the number of vehicles per kilometer and lane, and space, being the jam propagated to previous segments of the road. In this example, an adjacent road joins the road shown in this image causing "sudden" jams. Congestions become more and more serious since the vehicles of both roads meet in the same stretch. These data are used as inputs to the engine of congestion detection.

For test 1, a predefined tuned congestion level is surpassed and the system starts the process of congestion amelioration by means of messages to vehicles and road panels. For slow traffic, only vehicles registered in the service are warned. For more severe situations, the system generates notifications for all the vehicles involved and those driving previous stretches of the road that lead to the stretch with congestion. Vehicles that can take an alternative route could avoid the congestion and those that must pass (or decide to) pass through the congested stretch can adapt their speeds.

The consequences of launching services for congestion mitigation are shown in Fig. 7 in test 2. It can be seen how the severity of the congestion decreases, although congestion does not disappear completely. This is due to the limited number of alternative routes and the assumption that some drivers dismiss the warnings.

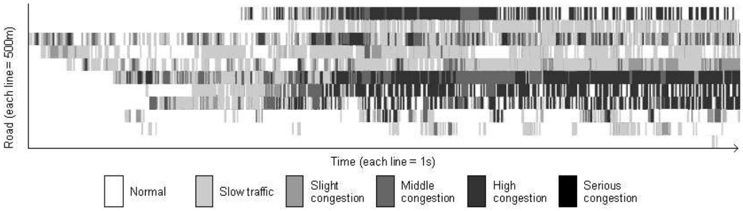


Fig. 7. Traffic density along time in test 2, after applying the system under consideration

7 Conclusions

A system that detects traffic congestions and suggests actions to mitigate them was presented. The system is based on events processing at different levels of abstraction, by means of CEP and ESP engines. In this architecture the events flow through the networks, saving costly storage of large amounts of data. After event processing, the system implementation in the traffic center may launch actions to influence the traffic conditions through informative panels or directly to the vehicles subscribed to the system. Simulations with thousands of vehicles show the benefits of this method in terms of reduction of traffic congestions on highway scenarios.

Acknowledgements

This research investigation has been carried out in the group of Intelligent Systems and Telematics of the University of Murcia with the support of the Spanish Ministry of Science and Innovation under the project SEISCIENTOS (TIN2008-06441-C02) and the Foundation Euromediterranean Institute of Water under the grant “Modelación hidrológica en zonas semiáridas” in the frame of Plan de Ciencia y Tecnología de la Regió de Murcia 2007–2010.

References

1. Luckham, D., Schulte, R.: Event Processing Glossary - Version 1.1 (2008)
2. Dunkel, J., Fernandez, A., Ortiz, R., Ossowski, S.: Event-Driven Architecture for Decision Support in Traffic Management Systems. In: Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems (2008)
3. Pawlowski, O., Dunkel, J., Bruns, R., Ossowski, S.: Applying Event Stream Processing on Traffic Problem Detection. In: 14th Portuguese Conference on Artificial Intelligence,EPIA (2009)
4. Open Geospatial Consortium (OGC), `http://www.opengeospatial.org/`
5. EspetTech: Esper Reference Documentation., `http://esper.codehaus.org`
6. Toledo, R., Zamora, M.A.: Collision Avoidance Support in Road with Lateral and Longitudinal Maneuver Prediction by Fusing GPS/IMU and Digital Maps. In: Transportation Research Part C: Emerging Technologies, vol. 18(4), pp. 611–625 (2010)

7. Toledo-Moreo, R., Zamora-Izquierdo, M.A., Ubeda-Miñarro, B., Gomez-Skarmeta, A.F.: High-Integrity IMM-EKF-Based Road Vehicle Navigation With Low-Cost GPS/SBAS/INS. IEEE Transactions on Intelligent Transportation Systems 8(3), 491–511 (2007)
8. Barshan, B., Durrant-Whyte, H.F.: Inertial Navigation Systems for Mobile Robots. IEEE Internatinal Transactions on Robotics and Automation II(3), 328–342 (1995)
9. Toledo, R., Sotomayor, C., Gomez-Skarmeta, A.F.: Quadrant: An Architecture Design for Intelligent Vehicle Services in Road Scenarios. Monograph on Advances in Transport Systems Telematics, 451–460 (2006)
10. Kokar, M.M., Matheus, C.J., Letkowski, J.A.: Association in Level 2 fusion. In: SPIE (2004)
11. Ceruti, M.G.: Ontology for Level-One Sensor Fusion and Knowledge Discovery. In: SPIE (2004)
12. Matheus, C.J., Kokar, M.M., Baclawski, K.: A Core Ontology for Situation Awareness. In: Proceedings of Sixth International Conference on Information Fusion, Cairns, Australia, pp. 545–552 (2003)
13. Ceruti, M.G., Kamel, M.N.: Preprocessing and Integration of Data from Multiple Sources for Knowledge Discovery. International Journal on Artificial Intelligence Tools 8(3), 159–177 (1999)
14. Waltz, E., Llinas, J.: Multisensor Data Fusion. Artech House, Boston (1990)
15. Dance, S., Caelli, T., Liu, Z.Q.: An architecture for a traffic scene interpretation system. Technical Report 94/12, Dept. of Computer Science. University of Melbourne (1994)
16. Jakobson, G., Lewis, L., Buford, J.: An Approach to Integrated Cognitive Fusion. Fusion Conference (2004), http://citeseer.ist.psu.edu/694904.html
17. Sun, T.Y., Tsai, S.J., Tseng, J.Y., Tseng, Y.C.: The Study on Intelligent Vehicle Collision-Avoidance System with Vision Perception and Fuzzy Decision Making, Intelligent Vehicles Symposium. In: Proceedings IEEE, pp. 112–117 (2005)
18. Ammoun, S., Nashashibi, F., Laurgeau, C.: Real-time crash avoidance system on crossroads based on 802.11 devices and GPS receivers. In: Proceedings of the IEEE ITSC 2006, Toronto, Canada, pp. 1023–1028. IEEE, Los Alamitos (2006)
19. Busch, M., Blackman, S.: Evaluation of IMM filtering for an air defense system application. Signal and Data Processing of Small Targets. In: Proc. SPIE, vol. 2561, pp. 435–447 (1995)
20. SUMO: Simulation of Urban Mobility., http://sumo.sourceforge.net

User-Friendly Robot Environment for Creation of Social Scenarios

Tino Lourens[1] and Emilia Barakova[2]

[1] TiViPE
Kanaaldijk ZW 11, Helmond, The Netherlands
tino@tivipe.com
[2] Eindhoven University of Technology
P.O. Box 513, Eindhoven, The Netherlands
e.i.barakova@tue.nl

Abstract. This paper proposes a user-friendly framework for designing robot behaviors by users with minimal understanding of programming. It is a step towards an end-user platform which is meant to be used by domain specialists for creating social scenarios, i.e. scenarios in which not high precision of movement is needed but frequent redesign of the robot behavior is a necessity. We show by a hand shaking experiment how convincing it is to construct robot behavior in this framework.

1 Introduction

A robot equipped with sensors and actuators is much more than a sensing, planning, and acting loop, as initially proposed in the 1980s, but merely a set of real time parallel processes that keeps this natural order of processing. The control software architecture used for these robots, i.e. the way to put these pieces together in a functioning system remains a challenging task. The present day software architectures need to cope with multiple processes running on different machines, often with different operating systems and hardware. Autonomous robots have the technical capacity to perform complex behaviors that can be beneficial to medical therapists, psychologists, designers, and educators. Despite of the potential of robotics in these areas, too much expertise is required today, and hence these robots have not yet been wideley used.

Our goal is to overcome this problem by making a programming architecture that facilitates easy construction of complex behaviors and interaction scenarios. We address 3 main issues from both technological and user perspectives. The first main issue is that even the most basic robot competency requires a vast amount of software. Therefore we developed an architecture that allows a high degree of re-usability of behaviors on all levels- from simple movement primitives to behaviors and scenarios. Second, we address the problem of parallel control over the different robot actuators that can be controlled by simple text-like commands. Third, since the major potential users of the robots are people with no or a moderate technical background, the architecture is developed towards an end-user platform, which implies usage of a graphical programming paradigm.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 212–221, 2011.

The robot architecture is developed as a part of TiViPE visual programming environment, which is a powerful integration tool.

As a particular application we feature training social skills to autistic children [1,2,3]. In a broader sense, educators, medical therapists, psychologists, scenario and storyboard designers, and software developers are the potential users of the architecture. Hence, we need to take a end-user approach towards robot scenario construction.

Several theories of controlling robots like the subsumption theory of Brooks [4], or a three layered approach of reaction-control, sequencing, and planning, emerged [5,6]. A recent development is the ROS open source operating system for robotics [7]. ROS runtime graph is a peer-to-peer network of processes that are loosely coupled using the ROS communication infrastructure. ROS is a is a distributed framework of processes that enables executables to be individually designed and loosely coupled at runtime, and aims to support multiple textual programming languages. Yet Another Robot Platform (YARP) also supports building a robot control system as a collection of programs communicating in a peer-to-peer way, its main focus is on long-term software development [8,9]. Another platform is the Open RObot COntrol Software (OROCOS) with focus on thread-safe and real time aspects between tasks [10]. Orca, initially part of the Orocos project, is an open-source framework for developing component-based robotic systems. It provides the means for defining and developing the building-blocks which can be connected to form arbitrarily complex robotic systems, from single vehicles to distributed sensor networks [11]. The authors of Orca make a comparison between seven different robotics software systems [12], but also address a key issue in robotics that *the sheer amount of software necessary for event the most basic competency is quite large*. These factors make software reuse attractive, i.e., *one hopes to integrate existing software modules within a software framework*. TiViPE which is a graphical programming environment [13], emphasizes on the integration of existing software routines from (existing) libraries without additional programming. TiViPE also covers the aspects of multiple processes on multiple (embedded) computers, peer-to-peer communication, graphical programming, massively parallel (GPU) processing, and multiple operating system support.

Because of the mentioned reasons, we build further on TiViPE visual programing environment.From the perspective of the user of such an environment, our focus will be to create a modular structure of command blocks, as already present in some simple solutions (Choreagraph) with the possibility to design new or redesign existing behaviors by a simple textual language used within the command blocks.

To satisfy this purpose, an incremental approach, that is similar to sticking Lego® bricks together, is used. It implies that the modular components serve as partial scenarios that can be modified by changing the order of usage, by replacing, or adding of such modules. Using such an approach will enable scenario designers, educators, and therapists to get full control over the robot.

Scenarios need to be easily adapted for user tests and rapid prototyping, and should allow for an incremental approach of scenario development. A graphical language will be used to realize these requirements. We use TiViPE to construct a robot architecture since it naturally can incorporate most of the aspects described in this section. In addition a substantial set of modules for sensory data processing is available within TiVIPE.

Section 2 describes a robot framework where a textual robot language is constructed with no more then 4 different control characters and 14 commands. This language provides full control over an advanced humanoid robot. Section 3 provides the robot sensory part. Most advanced robots provide audio and video data together with a set of values providing information about individual sensors. The following section describes how robot commands can be generated using and describe how the concept of using states is implemented using graphical components. Section 4 describes the work from the perspective of a user, gaining easy access to controlling a robot, without the need to have a background in robotics or software development. The paper ends with a summary and future work.

2 Robot Software Framework

The robot software framework includes the commands that interface with the robot, the language that insures non-conflicting operation of these commands by parallel actions of the robot and by multiple and conflicting action possibilities, and the visual programming environment that allows easy construction and visualization of the action stream.

2.1 Textual Language

A textual robot language insures that neither in depth knowledge of a robot hardware is required, nor the behavior designer is confined to a specific robot. A minimalistic language, such that the robot commands can be executed in a serial or parallel manner is the best choice. Therefore we choose for a language structure that resembles adding and multiplying of symbols, where the symbols are equivalent to behavioral primitives:

$$(a + b) * c + d * e \qquad (1)$$

Knowing that brackets bind stronger than multiplication, and multiplication has higher priority over addition, we can construct a powerful language that is in essence identical to the following statement from the formal logic:

$$[a \mid b] \ \& \ c \mid d \ \& \ e \qquad (2)$$

In terms of robots commands, a | b denotes that a and b are executed in parallel, d & e denotes that d and e are executed subsequently, first d then e, the square brackets denote the binding order of execution. Commands a, b, c, d, and e are

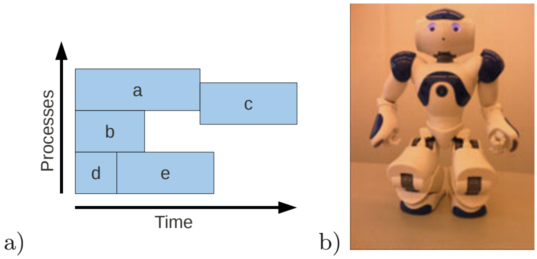


Fig. 1. a) Graphical representation of 5 commands as given in (2). b) Application platform - humanoid robot NAO.

individual elementary commands that can be executed on a robot. The graphical representation of (2) is given in Figure 1a.

These individual commands depend on the abilities of a robot, and might seem to contain a substantial number of commands that control the robot. However, even on a multi-functional humanoid robot like NAO, see Figure 1b, only few commands are needed to gain full control over the robot, as it will be explained in Section 2.2.

2.2 Robot Commands

Humanoid robot NAO can be controlled fully by as little as 7 different commands. However, for the ease of use, in our implementation we use 14 commands: two generic commands, two to enable or disable the LED's, two audio commands, and eight commands to get full control over all motors, as illustrated in Figure 2.

```
flush ()
wait (duration)ledset (name, value)
ledto (name, value, duration)
say (text[, volume[, language[, personality]]])
play (filename)
move (name, angle, duration[, stiffness])
movem (name, [angle, duration,]+ ...)
stiff ([stiffness, duration, idle,]+ ...)
walk (distance, duration)
walks (distance, duration)
walka (angle, radius, duration)
walkto (x, y, theta)
walkd (x, y, theta, frequency, duration)
```

Fig. 2. Commands that provide full control over a robot

Generic commands. Two generic commands `wait` and `flush` are available to enable the robot to synchronize its actions. The `wait` command lets the robot await (do nothing) for a given duration period. The `flush` command is used rarely to remove all the residual processes that are scheduled and not in execution, yet. While normal commands are appended, the `flush` command is executed immediately.

Commands for controlling the LEDs. Activating and disabling the LEDs is performed with the `ledset` command, in this case all leds are reached through a unique name, and can be given a value between 0 (`ledoff`) and 1 (`ledon`). For transition from the current color to a new color the `ledto` (also known as ledfade) command is used and a duration in milliseconds is provided to control the speed of change.

Audio commands. Two powerful audio commands `say` and `play` can be used to accomplish either text to speech conversion or to play an audio signal from a mp3 player engine.

Motor commands. The stiffness of a motor is usually set at the beginning using the `stiff` command. A typical setting is for example `stiff (1.0, 1000, 0)` that sets all motors to maximum (1.0) stiffness in 1 second (1000ms) time. The movement of a single motor is defined by addressing the motor, and specifying the desired angle of change over a specific time interval. For example `move (HeadYaw, 30.0, 1000)` moves the 'head yaw' motor to an angle of 30 degrees in one second time. In practice multiple moves for multiple motors are performed through a string-like command `[move (HeadYaw, 30.0, 1000) & move (HeadYaw, -30.0, 1000)] | [move (HeadPitch, 20.0, 700) & move (HeadPitch, -20.0, 600)]`. This command can also be written in a more compact manner like `movem (HeadYaw, 30.0, 1000, -30.0, 1000, HeadPitch, 20.0, 700, -20.0, 600)` by using the move multiple elements command.

3 Creating Behaviors with Incremental Complexity

The most simple robot behaviors are sequences of motor commands. Using sensory information requires specific design choices that bring another degree of complexity. Meta commands can combine behavioral sequences that are used often in a certain order. It may imply usage of states requires to meet a condition, before another behavioral t sequence can be started. Hybrid commands bring yet different practical concerns. The main design choices for each level of complexity is discussed in the upcoming subsections.

3.1 Using Sensory Information

Using sensory information is crucial for constructing reactive or more complex autonomous behaviors. However, it requires issues as how to construct parallel behaviors that have different cycle (or update) speeds. Moreover, there is a huge difference between the capacity of a robot to perceive (read sensory information) and act behaviors - the behavioral capacity is around two orders of magnitude smaller than the sensory capacity.

Behavior such as letting the robot walk requires already a sensory input, for instance, it needs to be sure that it is standing. In case the robot is lying down it first needs to get up. In order to decide which command to execute we need to get sensory information first.

To do so, a 'RobotSense' module is constructed in a way that it reads out all relevant sensory information from an a-priori known robot at a fixed, cyclic update speed, which is set by the user. Reading sensory information from a robot usually is a relatively trivial task, however, sensory information is obtained at different frequencies. Images in common can be obtained at a maximum rate of 30 frames per second, while internal data can be obtained at higher rates, for instance at 10 or 20 milliseconds rather than the 33 for video. Latter might form a bottleneck on bandwidth which might in turn lead to latency and a lower frame rate. Hence, audio, video and single valued information are decoupled, as illustrated in Figure 3.

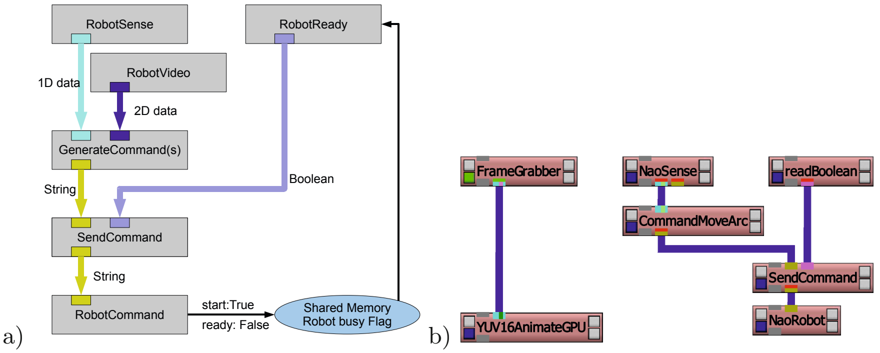


Fig. 3. a) Global processes and flows of a command generator and a robot module robot. b) TiViPE implementation of a) using a NAO robot and automated getting up before walking (in a circle).

A robot needs several seconds to get up, while less is needed for reading out sensory information. To avoid generating commands that the robot is not able to complete in time, a moderator process, called 'SendCommand' (see also Figure 3) is introduced. This process sends a command to the robot if all former commands are accomplished.

This so-called 'RobotCommand' (for NAO it is NaoRobot) processes the given commands. Switches to busy mode until it is ready signalling the 'readBoolean' module that its ready to receive.

This concept makes continuous sensing, and providing the latest command string on this basis possible without the need of specific knowledge on robot sensing or commanding. Hence, the focus is shifting to scenario's on basis of textual commands that are driven by internal information gathered over time and sensory data, hence one can focus on the 'GenerateCommands' type of modules.

3.2 Hybrid Command Structure

The textual robot language used provides an architectural advantage that the output of every command module is a string. Most of these modules will have a string as input as well, making it very easy to connect these modules. Since the language gives solely the option to connect new commands in a serial or parallel way string command generation is straight forward. It means that command generation becomes merely a choice of selecting a sequence of commands. The graphical tool allows multiple state sequences, combined with regular commands like 'Say', 'Wait', or its generic command 'GenerateCommand' that converge to a single string using the 'Serial' or 'Parallel' commands.

The graphical tool allows for merging modules to a single one, to get a layered abstraction for commands and command groups. On the other hand it is also possible to open and edit the compound structure of a merged module. The focus in the near future will be on the construction of these robot command modules on different levels of abstraction. Due to the modular concept of this approach an incremental abstraction and refinement of robot behavior will be constructed.

4 Example of Social Behavior Creation - User Point of View

In the previous sections a framework for parallel robot control with text-like commands has been given. Although simple from a technical point of view, the creation of robot behaviors has where a user solely needs to select graphical blocks and connect them such that they provide the right sensory or state information and generate a text string to command the robot. In case such a graphical block does not exist, the user needs to develop a routine call that evaluates relevant sensory information and generates a command string as described in Figure 2, keeping in mind that multiple commands can be placed between square brackets ([]), and are either separated by a `|` for parallel or a `&` for serial execution. Users are able to compound within TiVIPE a set of these blocks to a single block without additional programming.

The created framework allows a scenario designer to decide what blocks are needed and in collaboration with a developer to construct a set of useful graphical components of more complex robot behaviors or scenarios. When these blocks are available the users solely need to connect some of these modules, eventually these modules might have internal parameters that can be modified.

A student designer was used as a test person to see how the social scenario creation that includes robot implementation will be accomplished. He chose to make a hand shaking behavior between a robot and a child. He first got acquainted with the TiViPE and the robot embodiment. Subsequently, he explored the degrees of movement of the different joints of the robot, and got acquainted with the starting and end angles of joint movements. In parallel he did study on human motion organization - how each movement that is a part of a hand shaking interaction behavior is constructed and organized. For this purpose the student filmed different test persons performing the handshaking interaction and concluded that most of them would approach the partner, have a brief glance to the meeting points of the hands, and perform the handshaking.

To capture a realistic human movement, he used the following sequence of actions: (1)The usage of markers: markers were used to stick on actors arm; (2); The usage of coordinate: a 3-dimensional coordinate was been put to calculate the directions of each movement; (3) The usage of stick model: the stick model would also be used in this recording process, all human bodies of actors ware replaced by stick model; (4) The observation: the observation was taken by both video camera and monitor. The observation was mainly aimed to observe some other small changes during the hand shaking process, such as head movements, eye movements and body movements.

Based on this preparatory work, the handshake behavior was created within an hour. The handshake behavior contains 8 serial movem commands the third and fourth are as follows:

```
movem(RShoulderPitch, 33, 200, 33, 800, 32, 200,
      RElbowRoll,     38, 200, 35, 600, 36, 200, 35, 200,
      HeadPitch,      20, 800,
      HeadYaw,       -25, 800)
&
movem(RShoulderPitch, 33, 200, 33, 600, 30, 400,
      RElbowRoll,     36, 200, 36, 200, 35, 200, 32, 400, 33, 200,
      RHand,          57, 400, 20, 600,
      HeadPitch,      15, 600,  0, 600,
      HeadYaw,       -20, 600, -10, 600)
```

When completed, a user test of handshaking behavior was made in Maria elementary school in Helmond, The Netherlands. The children could naturally understand and perform the handshaking behavior without further guidance. Figure 4 shows some images from the user testing. Figure 4a shows the view from a side camera. Figure 4b and 4c depicts 2 snapshot views of the robot during handshaking interaction.

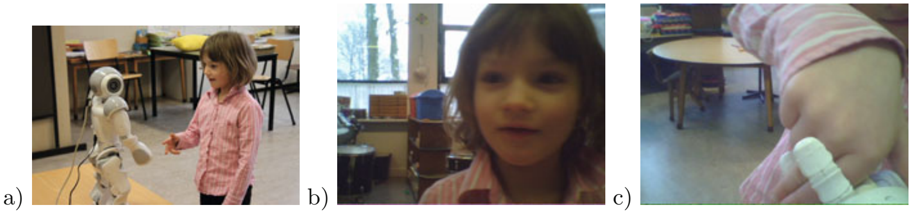


Fig. 4. a) Child shaking hand with robot. b-c) View from robot perspective.

5 Summary and Future Work

A key issue in robotics is that the amount of software necessary for even the most basic competency is quite large. Framework TiViPE has been extended with robotics library to substantially reduce the amount of coding necessary to control a robot.

The robot control platform is aimed to develop to an end-user platform. So far, a user-friendly architecture has been developed, which makes possible non-specialists to co-develop useful behaviors together with a person with some knowledge in programming. The future user of the robots is often not a programmer or a person with keen interest in robotics, hence we need to provide easy to use building blocks for these users to provide full control over a robot. This is done by providing ready graphical components for sensing the internal values of a robot as well as video and auditory modules. Further a single component is used to command the robot. No more than a single module for commanding a robot is required since a textual robotics language has been introduced with an a-priori known list of commands. This yields a simple but powerful language to gain full control over a robot. In this framework constructing scenario's has become an interplay between reading out sensory information and based upon this information generating a command string to control a robot. Creating these blocks have become entirely trivial, and in that respect we have shifted the complexity to creating scenario's where numerous blocks are involved.

So far the framework has been tested by design students and professionals that have created behaviors within TiVIPE [14,1] and image analysis for robot control [15,16]. Our future focus will be on scenario's and a layered component infrastructure, here the essence is to compound a set of blocks into meaningful partial sub-scenario's. Compounding itself is done without any addition programming with graphical programming environment TiViPE. Hence, once these sub-scenarios are created it will be easy for users to combine several sub-scenarios. For the particular application we are working on, we envision that therapists can do this to create training scenarios for autistic children. The scopes of application of such a generic framework, however, extends from exact control to applications in behavioral robotics and social robotics where issues of uncertainties need to be addressed.

Acknowledgments

We gratefully acknowledge the support of WikiTherapist project by the Innovation-Oriented Research Programme 'Integrated Product Creation and Realization (IOP IPCR)' of the Netherlands Ministry of Economic Affairs.

We also thank Ph.D. students, J. Gillesen and S. Boere, both supported by the WikiTherapist project, for developing novel robot behavior modules within TiViPE.

References

1. Barakova, E.I., Lourens, T.: Expressing and interpreting emotional movements in social games with robots. Personal and Ubiquitous Computing 14(5), 457–467 (2010)
2. Barakova, E.I., Gillessen, J., Feijs, L.: Social training of autistic children with interactive intelligent agents. Jounal of Integrative Neuroscience 8(1), 23–34 (2009)
3. Barakova, E.I., Chonnaparamutt, W.: Timing sensory integration for robot simulation of autistic behavior. IEEE Robotics and Automation Magazine 16(3), 51–58 (2009)
4. Brooks, R.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation 2(1), 14–23 (1986)
5. Firby, R.J.: Adaptive execution in complex dynamic worlds, Ph.D. thesis, Yale University (1989)
6. Gat, E.: Three-layer architectures. In: Artificial intelligence and mobile robots: case studies of successful robot systems, pp. 195–210. MIT Press, Cambridge (1998)
7. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
8. Metta, G., Fitzpatrick, P., Natale, L.: Yarp: Yet another robot platform. International Journal of Advanced Robotic Systems 3(1), 43–48 (2006)
9. Fitzpatrick, P., Metta, G., Natale, L.: Towards long-lived robot genes. Robotics and Autonomous Systems 56(1), 29–45 (2007)
10. Soetens, P.: A software framework for real-time and distributed robot and machine control, Ph.D. thesis, Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium (May 2006)
11. Brooks, A., Kaupp, T., Makarenko, A., Williams, S., Oreback, A.: Software Engineering for Experimental Robotics. In: Software Engineering for Experimental Robotics. Springer Tracts in Advanced Robotics, pp. 231–251 (2007)
12. Makarenko, A., Brooks, A., Kaupp, T.: On the benefits of making robotic software frameworks thin. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS (2007)
13. Lourens, T.: Tivipe-tino's visual programming environment. In: The 28th Annual International Computer Software & Applications Conference, IEEE COMPSAC 2004, pp. 10–15 (2004)
14. Lourens, T., van Berkel, R., Barakova, E.I.: Communicating emotions and mental states to robots in a real time parallel framework using laban movement analysis. Robotics and Autonomous Systems (2010), doi: 10.1016/j.robot, 08.006
15. Lourens, T., Barakova, E.: My sparring partner is a humanoid robot. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5602, pp. 344–352. Springer, Heidelberg (2009)
16. Lourens, T., Barakova, E.: Retrieving emotion from motion analysis: In a real time parallel framework for robots. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009. LNCS, vol. 5864, pp. 430–438. Springer, Heidelberg (2009)

Online Feature Weighting for Human Discrimination in a Person Following Robot

V. Alvarez-Santos[1], X.M. Pardo[1], R. Iglesias[1], A. Canedo-Rodriguez[1], and C.V. Regueiro[2]

[1] Department of Electronics and Computer Science
Universidade de Santiago de Compostela, Spain
victor.alvarez@usc.es
[2] Department of Electronics and Systems
Universidade da Coruña, Spain

Abstract. A robust and adaptive person-following behaviour is an important ability that most service robots must have to be able to face challenging illumination conditions, and crowded spaces of non-structured environments. In this paper, we propose a system which combines a laser based tracker with the support of a camera, acting as a discriminator between the target, and the other people present in the scene which might cause the laser tracker to fail. The discrimination is done using a online weighting of the feature space, based on the discriminability of each feature analysed.

Keywords: feature weighting, person follower, guide robot.

1 Introduction

Robots are expected to become part of our everyday's life, either as assistants, house appliances, collaborating in the care of elderly people, etc. Nevertheless, service robots will require sophisticated capabilities in order to be able to work cooperatively, side-by-side with humans in real world scenarios: one of these abilities is person-following. Service robots must be able to detect, track and follow people in order to interact with them in close proximity. Furthermore, these service robots will also have to be able to interpret human commands and react to them. This is part of the ongoing research under the project *intelligent and distributed control scenario for the fast and easy deployment of robots in diverse environments*, funded by the Spanish Government. Through this project we try to develop general purpose guide robots able to participate in different events providing useful information to the visitors. These general purpose guide robots and their control software must be developed in such a way that their deployment in different environments should not involve much prior effort or software development. In particular our general purpose guide robots must be able to go round providing useful information along routes that have been previously learnt. In fact to learn these routes our robot will have to follow a person along them. In this paper we describe a strategy to get a person-following behaviour that is robust, flexible to the environment and able to operate under difficult

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 222–232, 2011.

conditions such as dynamic, crowded and not structured environments. Our controller must achieve a high performance even under changing light conditions,or when there are people moving around and close to the person being followed (we will use the word *target* to refer to this person henceforth). Finally, the movement of the robot must be also safe, in the sense that it avoids colliding with other people or with obstacles placed in the environment [1].

The majority of the solutions previously published choose to track the target using either visual information[2] or data provided by a laser range scanner[1], although the most robust solutions use a combination of both sensors[3,4].

In this paper, we developed a controller which combines both sensors to achieve a person-following behaviour that fulfils two important characteristics: adaptability and robustness. Our controller will not have to be modified, or the parameters of the controller will not have to be tuned, to get the desired behaviour in different environments, or when the light conditions vary significantly. The tracking of the target is done using data from the laser, easy to get and process, and which offers a wide field of view, nevertheless the camera plays an irreplaceable role to discriminate amongst the target and other people present in the same scene (from now on we will call *distractors* to these people). Hence, the data provided by the camera will correct or help the laser tracker whenever the robot doesn't see person being followed for a short while, or when there are distractors around the target.

2 System Overview

Fig. 1 shows an overview of our system. In this figure, we can see that our system takes as inputs the images from a camera placed on the robot and data coming from a laser scanner. Our system will provide the relative position of the target with respect to the robot. We can notice the existence of two modules clearly differentiated: *the laser tracking module*, and *the camera discrimination module*. The first one uses the *leg detection* algorithm developed by Gockley et al. [1] to track the target. This algorithm combines the laser information with the last known position of the target. The camera discrimination module works on two clearly separated stages: first it uses the human detector presented by Dalal et al. [5], to extract the torsos of the people present in the scene. These torsos are the inputs of a target discrimination algorithm (second stage), which compares a model of the target with each torso in the image. As a result of this comparison each torso gets a dissimilarity measure. As we will describe later in this paper, this dissimilarity measure is achieved after an adaptive weighting process, that dynamically adjusts the importance of the different features used to distinguish the distractors from the target. The model of the target that is used to compare with the torsos detected in the image, is built during the first detection of the target, and it is dynamically adjusted while the robot moves. Whenever the camera module detects the target provides its position to the laser module which will check and correct its own estimation of the target's position. Finally, the position of the target is sent to a controller which, based on the

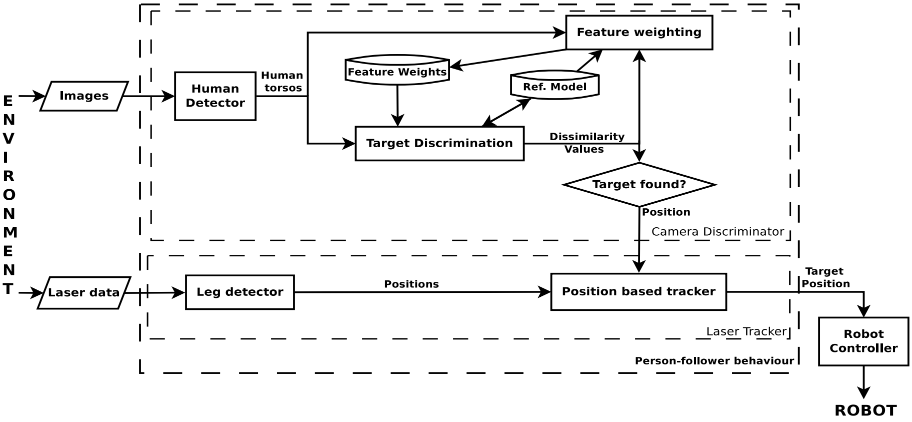


Fig. 1. Schematic representation of our system

concept of *potential fields*, will determine the motor commands so that the robot follows the target avoiding collisions with the environment.

We have noticed that the performance of our system is mainly due to the role of the camera module. Because of this, and also due to the fact that the laser module does not differ substantially from the one described in [1], we will only describe the camera module in detail in the next sections.

2.1 Target Model

Our system builds a model of the target that is composed of five normalized feature histograms. These histograms contain information about the colours and textures of the torso corresponding to the target. The colour is characterized by three components: hue (H), lightness (L) and saturation (S). The information about the texture is provided by the local binary patterns [6], and the edges detected with the *canny* method [7].

We used the HLS non-linear colour space since the linear ones, such as RGB, do not capture human intuitions about the topology of colours [8]. The HLS space is obtained by applying a non-linear transformation on the RGB space, thus obtaining a new base for colour representation: hue, lightness and saturation. Hue is the property of a colour that varies in passing from red to green, lightness is the property that varies in passing from black to white and saturation is the property of colour that varies in passing from red to pink. From these definitions, we can expect hue to be a good descriptor for torsos with vivid colours, while lightness will perform better on torsos that are almost white or black. Due to the difficult conditions of the environment: dynamic, crowded, changing light conditions,etc, we must consider each feature independently, in order to be able to dynamically select those which increase the discrimination amongst the target and the distractors.

Regarding the texture descriptors, the first one included in our feature space is a histogram representation of the local binary patterns (LBP). A local binary pattern describes the local shapes contained on a grey-scale image. Thus, in order to extract the LBP value corresponding to a pixel with a grey level g_c, we need to define a circle of radius R around that pixel, and from that circle select P equidistant points. The grey level of these P equidistant points will be used to get the LBP value:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c + a)2^p \tag{1}$$

where g_p are the grey levels of P points located in the aforementioned circle, a is constant value used to reduce the noise impact, and $s(x)$ is a thresholding function:

$$s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \tag{2}$$

In our case in particular we have set $P = 8$ and $R = 1$, in order to reduce the computational time required to calculate the LBP.

We have used a second texture descriptor aimed at discriminating the torsos upon their edge density. This gives a good measure of how textured an image is. This descriptor is based on the the edges of the image, obtained with the canny edge detector. Then, a convolution with a smoothing kernel is applied to include information about the separation among edges in the image. We would like to remark that this descriptor showed a high discrimination power during our tests.

Once the person-following behaviour starts, the target model is initialized starting from the person detected in front of the robot. We now will describe how we use the features we just described in order to find the target in the image.

2.2 Target Discrimination

The target discrimination algorithm pursuits the discrimination of the target from the distractors (other people in the scene different from the person being followed). This algorithm takes as inputs the torsos extracted from the people detected in the image, and outputs the dissimilarity of each torso with the target model.

In order to obtain the dissimilarity value amongst each torso and the target model, we need to get the histogram for the torso and each feature, $h_{torso,f}$, and compare it against the histogram obtained for the same feature but for the target model $h_{ref,f}$:

$$d_f = \sqrt{1 - \frac{1}{\sqrt{b^2 \bar{h}_{torso,f} \bar{h}_{ref,f}}} \sum_{i=0}^{b-1} \sqrt{h_{torso,f}(i) h_{ref,f}(i)}} \tag{3}$$

d_f is the normalized Bhattacharyya distance,$\bar{h}_{ref}$ and $\bar{h}$ are the average values of the histograms, b is the number of bins of the histograms. Starting from these distances we can compute the average dissimilarity amongst each torso and the target model:

$$dissimilarity = \frac{1}{n}\sum_{f=1}^{n} d_f \quad \in [0,1] \tag{4}$$

n is the number of features used to describe our target.

Every torso is compared with the target model to see if there is one that is similar enough to be considered the target (we use a threshold value $thr_1 = 0.4$ to do this). Nevertheless, if this dissimilarity is below a second and more restrictive threshold ($thr_2 = 0.2$), the target model will be updated using the torso identified as the target. This dual-threshold strategy avoids the pollution of the target histograms with wrong detections of the target. However, the dissimilarity measure described before (Eq. 4) is not yet robust enough to cope with real world conditions, such as strong illumination reflections, shadows, occlusions, and situations where the algorithm has to discriminate among similar torsos. Because of this, we have designed an adaptive weighting process to enhance the discriminability between the target and the distractors, this process is described in the following section.

2.3 Online Feature Weighting

Online feature weighting is the process of assigning high weights to the features that show a high discriminability between the target and the distractors. This is very useful when target and distractors share some common feature distributions. We can think, for example, in a target and several distractors wearing similar colour clothes, in this case it would be more useful to focus the dissimilarity measure on the texture features rather than in colour. This is what we try to achieve through the algorithm described in this section: an on-line adaptive process that changes the weights of the features trying to maximize the impact of the most discriminative ones.

As shown in Fig. 2, the feature weighting process consists on two tasks: a) building and updating a distractor list, and b) updating the values of the feature weights. The first task consists on building a distractor list containing all those torsos that have not been considered as the target. The distractors list is built and updated at every frame according to the following rules:

1. If a torso can be classified as the target, the rest of the torsos that have been detected in the same frame will be placed on the distractor list provided that they do not overlap in the image with the torso corresponding to the target.
2. If there is not a torso that can be classified as the target, those with a dissimilarity value higher than 0.5 will be put on the distractor list.
3. The list has a size limit of ten torsos. When the limit is reached, the oldest torsos will be removed. This size limit of the list is set to consider only the most recently seen torsos. A larger distractor list would save torsos which will not be seen again in a short period of time, thus reducing the performance of the feature weighting process.

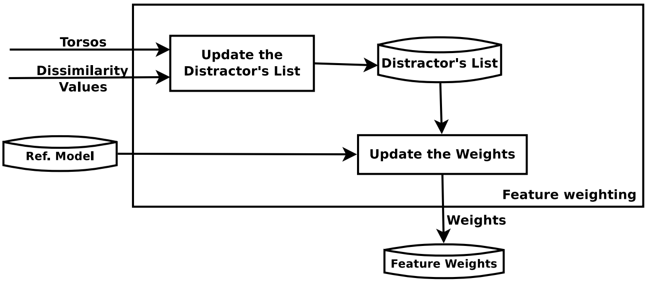


Fig. 2. Block diagram of the feature weighting process, which maintains an internal distractors list in order to decide which features present the highest discriminability

Once the distractor list has been updated, it is used to evaluate the discriminability of each feature. To do so each feature will be given a score by adapting the concept of *peak difference* [9] to our problem. This score is defined by:

$$score_f = averageDissimilarity_f - lastTargetDissimilarity_f \tag{5}$$

where $score_f$ is the score given to feature f, $lastTargetDissimilarity_f$ is the dissimilarity amongst the last torso accepted as target and the target model when this happened. Finally, $averageDissimilarity_f$ is the average distractor's dissimilarity for feature f, calculated using the distractor list:

$$averageDissimilarity_f = \sum_{i=0}^{s-1} \frac{d_f}{s} \tag{6}$$

where s is the size of the distractors list. d_f is the Bhattacharyya distance (Eq. 3) between the histograms of feature f for the target and the i-distractor on the list. Finally we will consider the scores previously described to determine the weight of each feature:

$$w_f = \begin{cases} 0 & \text{if } score_f < 0.25 \\ 0.5 w_f + score_f & \text{if } score_f \geq 0.25 \end{cases} \tag{7}$$

where w_f is the weight for feature f. After Eq. 7 has been applied, it is necessary to apply a normalization so that the sum of all the feature-weights is equal to one. We can notice that equation 7 sets the weight value to zero when the score of a feature is less than 0.25 (what means that the discriminability of that feature is poor and therefore it should not be used to compute the dissimilarity values Eq. 4). Finally, when all the features have a zero score, due to their low discrimination ability, the weight of the feature with the highest score will be set to one, leaving the rest of the weights with a zero value.

Finally, we have to modify the dissimilarity equation (Eq. 4) to include the feature weights just described:

$$weightedDissimilarity = \sum_{f=1}^{n} w_f d_f \tag{8}$$

3 Experimental Results

We have tested our proposal over two video sequences captured while the robot is performing the person-following behaviour and when there are several distractors trying to make the system fail. These videos were logged in the Department of Electronics & Computer Science, at the University of Santiago de Compostela, Spain, and they can be seen in (`http://www-gsi.dec.usc.es/ roberto/TIN2009.html`). Because of the position of the lights inside the building we will see the performance of our system when the illumination conditions change significantly. This alters the colour aspect of the torsos detected from the camera located on the robot. Texture is also affected by both illumination and person's pose changes. The robot used in the tests is a Pioneer 3-DX, equipped with a Sick LMS 200 laser scanner and a camera, both, the robot and the camera are controlled from a laptop with a 2.4GHz Core 2 Duo processor. The system was coded using the OpenCV library [10] and C++ language, our software is able to process about 10 frames per second (320 × 240 images). We compared the performance of the camera discrimination module using both, the classic average dissimilarity values Eq.4, and our adaptive weighted dissimilarity Eq. 8.

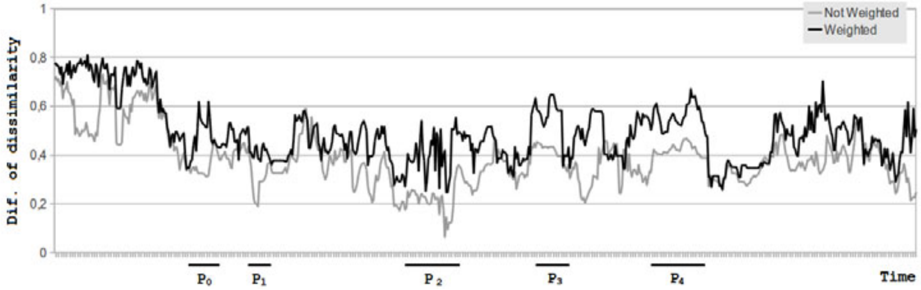


Fig. 3. Evolution of the average dissimilarity between the distractors on the *distractors list* and the target. We can see the results using the classic average dissimilarity and the online feature weighting.

In the first video sequence there are two distractors moving close to the target being followed by the robot. Figure 3 shows how the performance increases when feature weighting is used. On average there is an increase of around 29% on target discriminability (the dissimilarity between the target and the distractors increases from 0.38 to 0.49). Fig.4 illustrates how the weight values are dynamically adapted as the time goes by, this change of the weights is mostly due to the varying illumination conditions. In general the situation in which there are several distractors moving close to the target is the riskiest. Luckily, when there is only one distractor (even thought the distractor might change after several frames), the weighted dissimilarity increases the separability between it and the target (regions P_0,P_3 and P_4 in Fig.3 and Fig.4). It is also noticeable that when

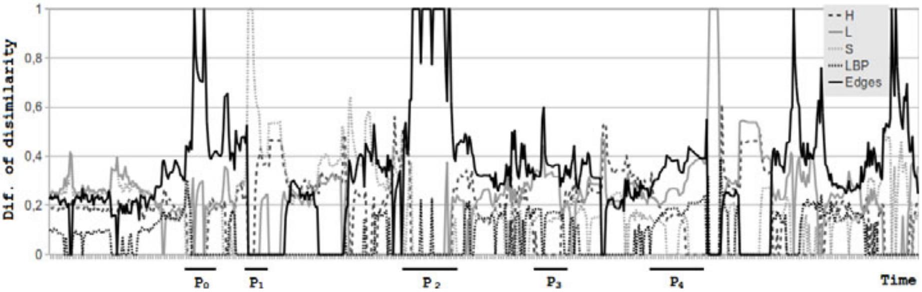


Fig. 4. Evolution of feature weights during first video sequence

class separability is very low, one or two features are chosen while the rest of them are discarded (region P_1 and P_2 in Fig.3 and Fig.4). Table 1 shows the confusion matrix which shows the recognition ratio, and the number of false negatives and positives. The results are shown when using our proposal and the classic average weighting. In order to get the confusion matrix we processed all the video frames apart from those where the target was standing in front of the robot without moving, or when there were no illumination changes or no distractors. Despite of the fact that the target recognition seems to be worse with our system than with the classic weighting approach, the number of false positives is clearly lower (situations in which the distractor is misclassified as target). This is something relevant since missing the target during some few frames will not alter the behaviour of the robot too much thanks to the laser module, nevertheless mistaking the target for the distractor would cause an important misbehaviour on the robot.

Table 1. Confusion matrix of the first video sequence. Results are shown using the classic approach (Left), and using our proposed weighting of the feature space (Right).

Actual\Classif.	Target	Distractor
Target	215(95,5%)	10(4.5%)
Distractor	16(2,8%)	544(97%)

Actual\Classif.	Target	Distractor
Target	189(84%)	36(16%)
Distractor	0(0%)	560(100%)

Regarding the second video sequence, it shows the movement of the target and one distractor. Fig. 5 shows the increase of the performance when feature weighting is used. In this second case the dissimilarity is increased, on average, from 0.35 to 0.58 (66% increase on target discriminability). Neither the target nor the distractor wear vivid colour clothes, therefore texture features and lightness are expected to be the main discriminators. Figure 6 shows that the results confirm what was expected: two of the total number of features represent 80% of the total sum of weights during most of the video sequence. Analysing Figure 6 we can notice that there is a small time interval at the beginning (from T_0 to T_1), in which the weight values are similar to each other and the dissimilarity between target and distractor is not very high, this is due to the illumination conditions and pose of the target. Nevertheless this situation changes soon, achieving a small set of prominent features that increase the separability amongst target and distractor. As we can see in Table 2, the recognition ratio is clearly superior with our system than with the classic average weighting, and once again the false positive ratio has been reduced.

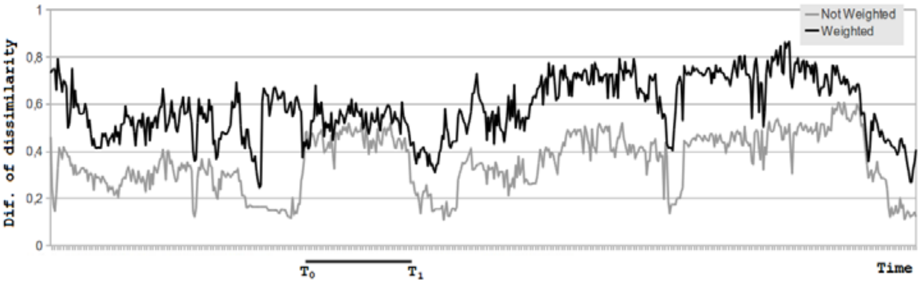


Fig. 5. Evolution of the average dissimilarity between distractors included in the *distractors list* and the target during the second video sequence.

Table 2. Confusion matrix of the second video sequence. Results are shown using the classic approach (Left), and using our proposed weighting of the feature space (Right).

Actual\Classif.	Target	Distractor
Target	139(80.4%)	34(19.6%)
Distractor	8(3,6%)	212(96,3%)

Actual\Classif.	Target	Distractor
Target	142(99.3%)	1(0.6%)
Distractor	3(1.3%)	217(98.6%)

These results confirm that our system is capable of adapting to difficult conditions through an on-line weighting process able to maximize the dissimilarity target-distractor. We have also tested the person-following controller operating on the real robot during several 10 minute walks around the corridors and the hall of our department. The robot had to follow the target when both the corridors and the premises of the building were usually crowded with students walking around. The robot's maximum speed was set to 1 m/s, thus allowing the target to walk at normal speed. The robot was able to avoid collisions with the environment thanks to a potential field method implemented on the robot controller. The robot was able to follow the target keeping a distance that ranged from 0.5 meters to 6 meters, although the average distance between the robot and the target was two meters.

4 Conclusions

We have described a person-following behaviour which combines a laser based tracker, with the discrimination power of a camera. The camera uses a novel online feature weighting method. This method evaluates the discriminability of

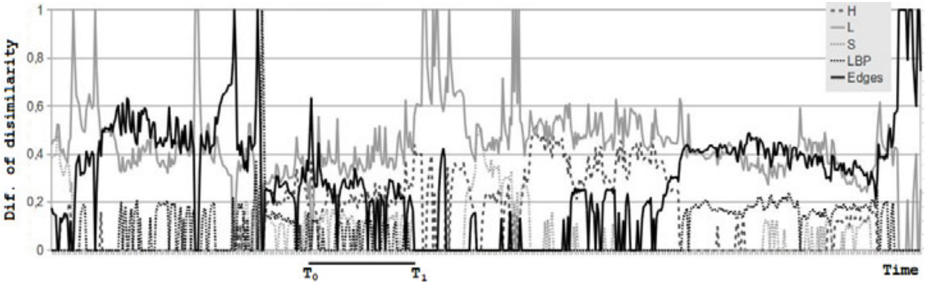


Fig. 6. Evolution of feature weights during the second sequence

each feature included in the feature space and enhances the weight according to the score obtained by each feature. This increases the dissimilarity between the target being followed by the robot, and the distractors moving close to it. The experimental results show the high performance of our proposal working on a real robot. The person-following behaviour was also tested during large periods of time on crowded environments showing the expected behaviour.

Acknowledgements

This work was supported by the research grants TIN2009-07737 and INCITE08PXIB262202PR.

References

1. Gockley, R., Forlizzi, J., Simmons, R.: Natural person-following behaviour for social robots. In: Proceedings of the ACM/IEEE International Conference on Human-robot Interaction, pp. 17–24 (2007)
2. Yoshimi, T., Nishiyama, M., Sonoura, T., Nakamoto, H., Tokura, S., Sato, H., Ozaki, F., Matsuhira, N., Mizoguchi, H.: Development of a person following robot with vision based target detection. In:IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5286–5291 (2007)
3. Bellotto, N., Hu, H.: People tracking and identification with a mobile robot. In: International Conference on Mechatronics and Automation, ICMA 2007, pp. 3565–3570 (2007)
4. Bellotto, N., Hu, H.: Multisensor-based human detection and tracking for mobile service robots. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39(1), 167–181 (2009)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2005, vol. 1, pp. 886–893 (2005)
6. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29(1), 51–59 (1996)

7. Canny, J.: A computational approach to edge detection. In: Readings in computer vision: issues, problems, principles, and paradigms, vol. 184 Morgan Kaufmann Publishers Inc., San Francisco (1987)
8. Forsyth, D.A., Ponce, J.: Computer vision: a modern approach. In: Prentice Hall Professional Technical Reference (2002)
9. Collins, R.T., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1631–1643 (2005)
10. Bradski, G., Kaehler, A.: Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Sebastopol (2008)

Improving Area Center Robot Navigation Using a Novel Range Scan Segmentation Method

José Manuel Cuadra Troncoso[1], José Ramón Álvarez-Sánchez[1],
Félix de la Paz López[1], and Antonio Fernández-Caballero[2]

[1] Dpto. de Inteligencia Artificial - UNED - Madrid, Spain
{jmcuadra,jras,delapaz}@dia.uned.es
[2] Departamento de Sistemas Informáticos-UCLM-Albacete
caballer@dsi.uclm.es

Abstract. When using raw 2D range measures to delimit the border for the free area sensed by a robot, the noise makes the sensor to yield a cloud of points, which is an imprecise border. This vagueness pose some problems for robot navigation using area center methods, due to free area split points locations. The basic method, when locating split points, does not take into account environmental features, only the raw cloud of points. In order to determine accurately such environmental features we use a novel range scan segmentation method. This method has the interesting characteristic of being adaptive to environment noise, in the sense that we do not need to fix noise standard deviation, even different areas of the same scan can have different deviations, e. g. a wall besides a hedge. Procedure execution time is in the order of milliseconds for modern processors. Information about interesting navigational features is used to improve area center navigation by means of determining safer split points and developing the idea of dynamic split point. A dynamic split point change its position to a new feature if this new feature is considered more dangerous than the one marked by the split point.

1 Introduction

In range sensor measurements noise sources may be very disparate, like: intrinsic sensor error, sensor calibration errors and the environment itself. The noise portion due to the environment cannot be neglected, but it can be the main noise source. For example, laser measures over a garden hedge carry much more error, in standard deviation terms, than measures over a plain wall. Although devices technical specifications usually carry information about intrinsic device noise and calibration procedures, we have no prior information about environmental noise, so we need to make assumptions about its behavior. If we want to develop a procedure being able to handle worlds with surfaces of different characteristics at a time we have to reduce or to relax our assumptions about noise. As more general our noise model as more flexible and robust navigation we get.

Filtering noise we can get an ideal representation of the world as a set of line segments and curved segments, this procedure is known as scan segmentation or

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 233–245, 2011.

line extraction. In section 2 we give a brief review of the subject. The proposed line extraction procedure is explained in section 3 through section 5.

In section 3 a sequential line estimation procedure is presented. Using directly the line equation in polar coordinates, we have developed an Extended Kalman Filter (EKF) for sequential line estimation, being noise variance unknown and depending on measured distance. Our filter formulation is equivalent to regression models formulation, so we get all the statistical properties, test for hypothesis contrast, etc., known for regression models theory.

Kalman filters need initial values for their parameters, the only place for collecting this information are the measures. A set measures, called seed, is used to get an initial estimation for the filter parameters, a clustering procedure, section 4, is used to determine good seeds positions. This clustering procedure is based in scale-space techniques, we have develop a statistic filter to reduce the number of contour curves due to noise, making the procedure more reliable and efficient.

In section 5 the designed EKF ability is used for detecting outliers to detect segments ends, overlapped segments compete for common measures and finally, due to filter statistical properties, test from model regression theory to merge similar segments are applied.

Section 6 presents some uses of segmentation results for making robot navigation safer when using center of area methods: better split points location and moveable split points.

2 Line Extraction Review

It is not intended in this section to make a detailed review of the subject (interested reader is referred to [9]) but just to highlight the aspects directly motivating our work or more related to it. First question attracting our attention was the existence of distance thresholds and the second one was noise effects and models.

Distance thresholds are usual criteria to determine if a point belongs to a segment based on distance between points. Some known methods using distance threshold are: Split and Merge [5], Iterative End Point Fit [4], Successive Edge Following [3], Random Sample Consensus Algorithm (RANSAC) [6] and Line Tracking [14]. Our objections to distance thresholds are two. First, taking measures radially there is no limit, theoretically, for distance between two consecutive segment points, it depends on incidence angle. Second, points belonging to non concurrent segments could be misclassified if they are close to each other.

Second question to consider is noise models, specially in probabilistic methods. When noise variance is used in the noise model, it is supposed to be constant and known, in section 1 we commented the possible big influence of the reflexion surface on the measure error. We have not found comments in literature mentioning explicitly the possibility for unknown, changing or high variances, but we can give some references where fixed known variance is theoretically required [18], or used in experiments [4,9]. Normally this variance is the value found in device technical specifications plus some constant.

Our work use Kalman filter (KF) estimation as in [18,4,16]. Also it uses a clustering procedure prior to line estimation, this procedure is used in several works cited previously. Clustering provides a representation of the world at a coarse scale, the problem is to select the proper scale level to represent it in an adequate way. This question leads to the concept of natural (most significant) scales [12]. The selected method has been Interval Tree by Witkin [17], this is the pioneer paper about scale-space. Scale space theory [8] provides a multi-scale representation of a signal formed by smoothed versions of the signal. Smoothing is performed convoluting the signal with gaussian kernels with different variances, the changing scale parameter. Features are normally discovered as extrema or inflexion points of the signal, thus they are represented as zero-crossing level curves in the scale-space representation, see fig. 1 in section 4. Witkin's Interval Tree procedure associates a rectangle to each curve and defines a stability criterion for rectangles, most stable rectangles define natural scales.

3 Line Estimation

3.1 Measures Model

The problem about using rectangular coordinates with noisy measures taken radially is that both variables, x and y, are random and correlated [7], and there is not a non-stochastic independent variable. In this case many of the properties and results from regression theory. The only case where stochastic independent variables are allowed is when they have to take equally spaced values. This is the case for angles in a laser scan, so polar coordinates are an adequate statistical frame for line estimation from laser measures.

The equation of a line, not crossing the coordinates origin (pole), in polar coordinates is:

$$r = f(\varphi) = \frac{\rho}{\cos(\varphi - \theta)} \qquad -\frac{\pi}{2} < \varphi - \theta < \frac{\pi}{2} \tag{1}$$

being r the measured distance, φ the angle of the measure, ρ line distance to pole, and θ angle between x axis and the line.

Our measure model is: $r_n = E[r_n] + \varepsilon_n$, being $E[r_n]$ the expected value of the measure and ε_n the noise. Noise has a gaussian probability distribution with zero mean and variance $\sigma_n^2 \propto h^2(E[r_n])$, being h a known function. In our experiments we use a noise standard deviation proportional to the expected measure, so $\sigma_n^2 = \sigma^2 E^2[r_n]$, being σ an unknown constant representing the deviation per unit measure, or roughly the noise proportion.

3.2 EKF for Line Estimation in Polar Coordinates

As we commented in section 2, in order to detect outliers it is necessary to estimate the line sequentially and to have a test for outliers detection. Sequential linear regression estimation studied by Plackett in the 50's [11]. Later Peña [10], saw the equivalence between Plackett formulation and KF formulation, see eq. 2 for EKF formulation, so every result, property, test, etc., from regression theory is applicable. For example KF formulation is independent of σ^2 and prediction test from regression theory may be used as a outlier detection test. He gave a procedure to sequentially compute the residual sum of squares (RSS), needed for σ^2 estimation, reaching a linear temporal complexity for the process.

To deal with non-constant variances weighted least squares may be used [7]. When $\sigma_n^2 = \sigma^2 h^2 \left(E[y_n]\right)$, being y_n the independent variable and h a known function, a two rounds procedure can be applied due to expected values are usually unknown. Firstly, ordinary least squares, over the whole data set, are applied to obtain estimated measures $\hat{y}_i$ and a second least squares round, over the whole data set, is applied weighting data using $w_i = w(\hat{y}_i) = \frac{1}{|h(\hat{y}_i)|}$. Using Peña's KF formulation and filter independence of σ^2, matrix covariance noise remains $\frac{1}{w_i^2}$, so data have not to be weighted if this matrix is used. In this work we are going apply the two rounds procedure using KF, but rounds just involve the new incoming data.

During initial experiments, with linear models, we observed that if weights were significantly less than 1 at initial stages could lead to a bad result. In these cases Kalman gains were too small, making filter response to innovation small too. The solution was to scale weights, by a factor ω, in order to make initial weight values near or over 1, for example, $\omega = \max\left(|h(y_s)|\right)$, being y_s the seed. Estimation of parameters and their covariances matrix are unaffected by this scale factor. Noise deviation estimation is scaled by the same factor.

Equation 1 represents a non-linear model, so we need an EKF. The EKF formulation, adapted from [10], is simply:

$$\begin{aligned} \beta_n &= \beta_{n-1} \\ r_n &= \frac{\rho_n}{\cos(\varphi_n - \theta_n)} + \varepsilon_n \end{aligned} \tag{2}$$

first equation is the state equation and second one is the measure equation. The state parameters are $\beta = \begin{pmatrix} \rho \\ \theta \end{pmatrix}$ and ε_n is gaussian distributed with zero mean and covariances matrix $\sigma_n^2 = \sigma^2 h^2 \left(E[r_n]\right)$.

4 Clustering

The scale-space procedure, described in section 2, can be applied for range scan segmentation. Fig. 3 a) shows a simulated world composed by plain walls and a hedge, where noise was generated following the model explained in subsection 3.1 with $w_i = |h(\hat{r}_i)| = \frac{1}{\hat{r}_i}$, so noise is proportional to measure. Measures on the walls have a standard deviation proportional to the measure being of $2\,cm.$ at $10\,m.$, standard deviation is 10 times higher on hedge.

In fig. 2 we can see super-imposed the polar representation of that world. Segments edges are extrema and inflexions points of the curve, so first and second derivative can be used to find them. Zero-crossing level curves for first and second derivative of the scale-space surface are shown in fig. 1. Specially for the second derivative noise yields a lot of curves. This curves could magnify low level environmental features yielding a too fragmented world representation. An adequate selection of scale levels, specially the lowest level could remove eventually the majority of the small curves. We propose a filtering procedure that can adapt itself to the mean amount of environmental noise, removing noise.

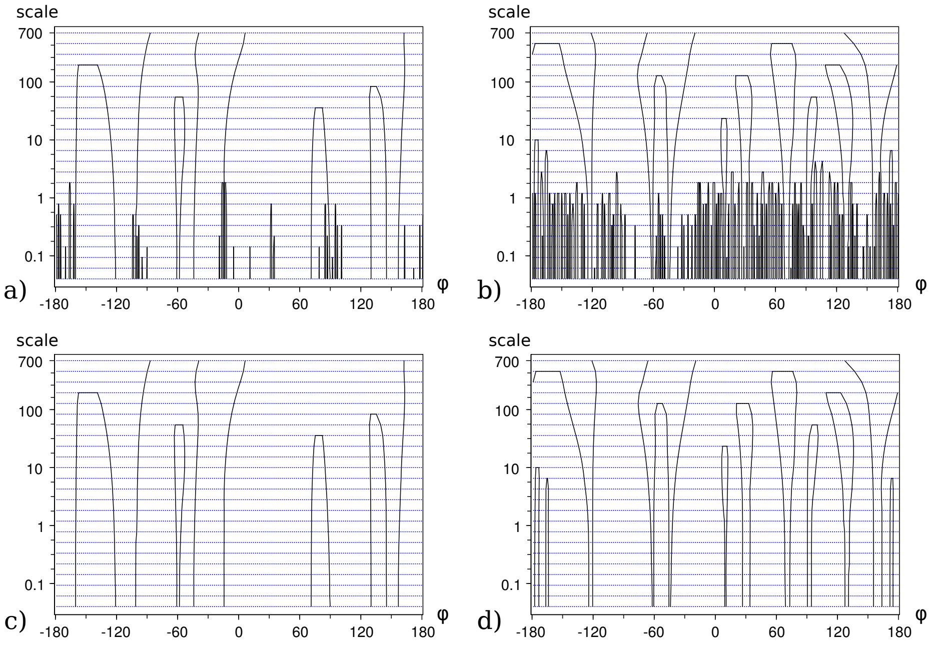


Fig. 1. Zero-crossing level curves for first, a) and c), and second derivatives, b) and d), of scale-space surface. In c) and d) shown the curves remaining after noise filtering. Horizontal blue lines are discretized scale-space parameter values.

The filtering procedure is based on a characterization for the curves generated by noise. In fig. 1 a) and b) we can appreciate that noise generated curves are usually significantly narrower and lower than curves generated by high level features, these are most of curves showed in 1 c) and d). So we looked for a basic characterization based on areas under curves. Experiments using fixed and simple patterns of noise gave an approximate interval for the maximum area of noise generated curves that roughly separates both types of curves. Values for noise deviation in [0, 0.05] were used, this upper ends represents 50 cm. at 10 m. We have choose inside that interval the value 50 as area upper threshold. Curves with area under the threshold are preliminary classified as noise generated. To refine the search for noise generated curves we compute the mean $\bar{h}$, and the standard deviation s_h for selected curves heights, setting $\bar{h}+ks_d$, as the definitive upper threshold, usual values for k are between 2 and 3. The value $\bar{h}$ is used as lowest level for the Interval Tree procedure, see section 2.

The results of the filtering procedure are shown in fig. 1 c) and d), only some noise generated curves remain. These curve are located at the hedge segment where the noise is significantly higher than in the rest. After filtering, the Interval Tree procedure is performed in order to find the most stable environmental features, results are shown in fig. 2.

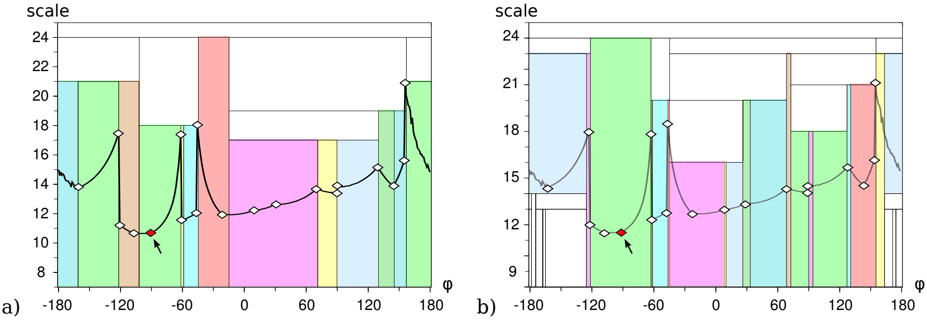


Fig. 2. Stable features, colored rectangles, discovered from level curves after filtering. a) First derivative and b) second one. The difference between lower limits in y axis is due to adaptation to noise, second derivative enhances noise effects. World representation in polar coordinates has been superimposed. No features generates by noise has been discovered and only one real feature has been missed, it is marked with an arrow at $-90°$.

Features locations discovered using both derivatives have to be merged, nearby features a mixed in one and groups of missing measures are taking in account using their ends as features. Isolated missing values are interpolated and groups are replaced with convenience noiseless values. Finally, the discovered features are considered as fuzzy ends of segments, then sets of consecutive measures are extracted from each segment core to obtain the seeds for KF initial values estimation.

5 Merging

Applying the clustering procedure explained in section 4 we get a set of seeds to obtain initial estimation for KF bootstrapping. In order to achieve a balanced estimation we have to introduce new points in the filter consecutively at both seed ends. Experiments showed a significant reduction in estimation errors maxima when doing balanced estimation.

The selection of an adequate number of measures to include in the seeds poses two questions. First, as longer the seed the smaller the initial error estimation is. Second, as longer the seed, the lesser the number of points we can test for outlier detection. Take into account that usually test critic values are high if sample is very small, in our case we needed a least 6 measures. Intensive simulations showed that 9 or more measures give a good accuracy, even 5 are not bad. So we need at least 15 measures to start a reliable outlier detection. This number looked to high to us so we studied the possibility for including seed points in KF procedure.

Reusing the seed carries the problem of statistical dependence, thus we considered to obtain the initial estimations from the seed but using a different regression framework other than least squares and, then, to test experimentally the effects. For initial estimation the Repeated Median [15] was chosen, in median based regression the target is to minimize the sum of absolute residuals instead of the squares. The experiments shown a small increment in final estimation errors when including the seed but also a significant increment in the ability for early outliers detection. So we propose to include seeds in KF procedure using median based regression for initial estimations.

Before starting KF estimation procedure a condition for segment end detection has to be proposed. Two types of outliers exist: proper outliers, points belonging to the segment but a bit far from it, and improper outliers, points belonging to other segments. Finding only one outlier as segment end detection is not enough, but waiting for two consecutive outliers is too restricitive and can lead to failing to detect many connected segments ends when noise variance is high. As an intermediate solution our proposed end detection condition is: two outliers in two or three consecutive measures, so that if α is the significance level for outlier detection, then, $2\alpha^2$ is, approximately, the significance level for end detection.

In order to stop estimation, segments ends have to be found at both seed sides. We had to impose two more conditions for estimation stop. First, if process has reached the beginning of the previous seed or the end of the next one, estimation is stopped at the corresponding seed side. These stop points are set in order to avoid excessive overlapping between connected segments as result of high noise levels. Second, if the estimated noise standard deviation is higher than a limit, the filter is considered as out of control, the limit was set to 0.1, equivalent to $1\,m.$ at $10\,m.$

When all KF estimations are concluded, a search for groups of points not belonging to any segment, holes, is performed. One new segment is estimated per each hole. At this point, overlaps between segments may exist, so we need a criterion for assigning a group of consecutive points to each segment. Comparing noise variance estimation for each segment, when the point was added to the estimation procedure, was the most accurate criterion for resolving overlaps we have found. Sign of difference between variances at overlap center is computed then the first change in difference sign is searched at both center sides, if it is found, the overlap is split, if not, one of the segments gets the overlap. When overlaps are resolved a new search for holes is performed, but in this case estimation are restricted to holes limits.

Specially when noise is high, clustering procedure could yield more than one seed for a "true" segment, so we need a procedure for merging similar segments. We have selected Chow's test [7], from regression theory, designed for comparing

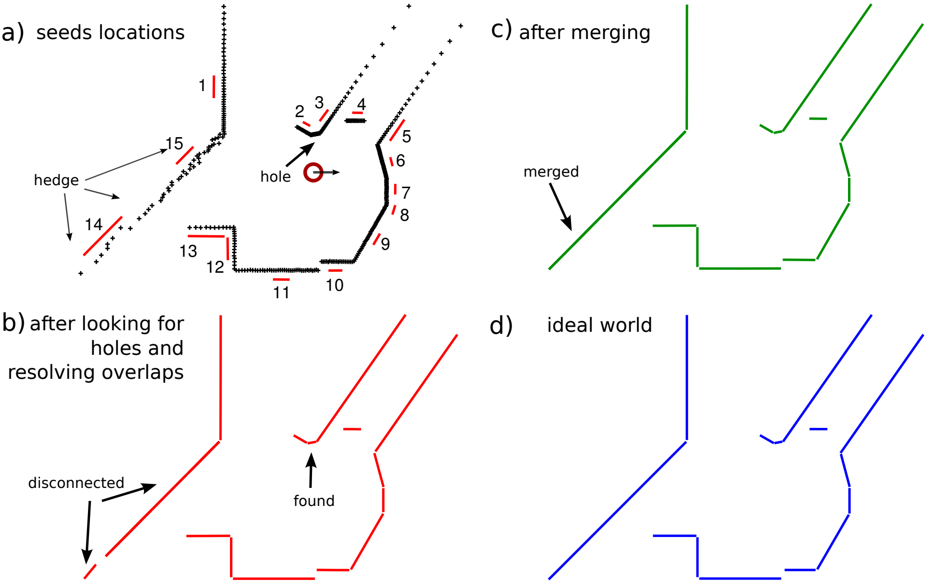


Fig. 3. Pictures sequence showing the application of the estimation and merging procedure described in section 5. World is composed by plain walls and one hedge where the noise is 10 times higher, world dimensions are 17.5 m × 12.5 m. Red(black) circle in a) represents robot and the arrow at robot center indicates orientation $\varphi = 0$. A picture of the ideal world used to generate the measures is shown for comparison purposes.

parameters of regressions over different sets of data. In fig. 3 an example of the whole process is shown.

To analyze the performance on a computer a preliminary C++ implementation of the procedure, yet in development, was tested on laptop with a Core Duo 2 processor at 2.28 Ghz. New robots are being shipped with similar processors and several well known old models can be now upgraded. For 181 measures with noise standard deviation $\sigma = 0.002$, $2\,cm.$ at $10\,m.$ the procedure takes about 2.5 milliseconds, for 360 about 6 ms. and for 720 about 9.5 ms., these times suggest a complexity near linear order. For $\sigma = 0.02$, times increase about $25 - 30\%$. Taken into account that an usual laser sampling period takes 100 ms or longer, proposed procedure execution only takes a bit of this time.

6 Improving Area Center Navigation

The basic area center method for robot navigation was proposed previously in [1,2,13]. Basically the robot follows the center of area of its perceived free area while it is accessible. Robot velocity vector is proportional to distance of area center in robot local coordinates. A basic concept in area center navigation is the split points, where the perceived free area is split in two sectors when area center becomes inaccessible and one of them is selected to follow its area center. Successive split points can be set, while selected sector aperture is wide enough, allowing robot to pass across narrow passages and to drive in crowded environments. For goal reaching a virtual split point can used to select a sector in goal direction.

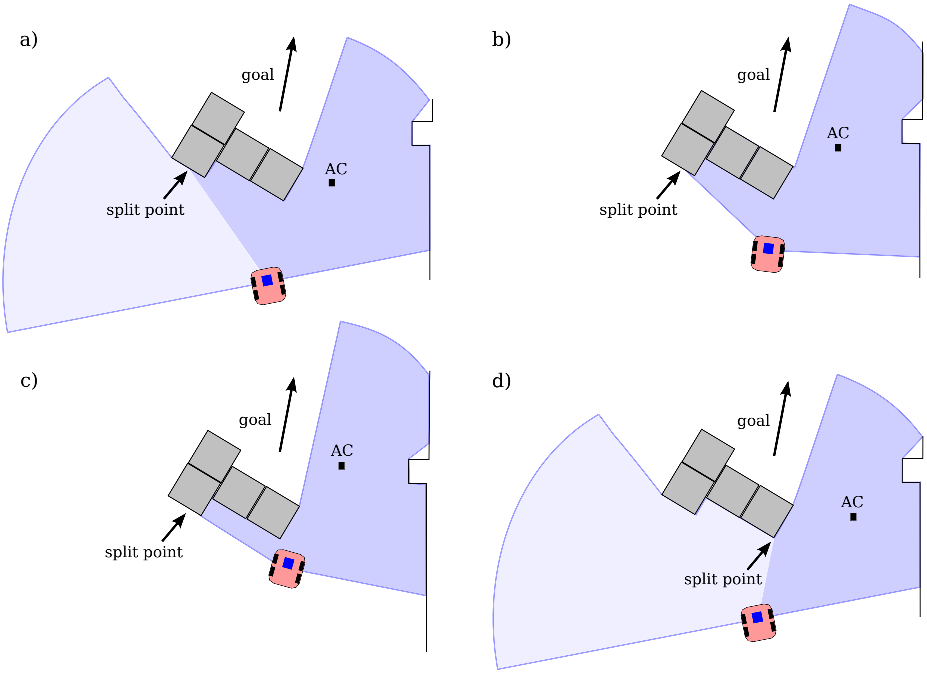


Fig. 4. Bad split point selection example case. In a) the center of area for the full free area becomes inaccessible in the point marked with an arrow and it becomes the split point. Due to goal direction right sector is selected and robot follows the new area center marked with AC b). But area center path is not enough far away from obstacle leading robot to a crash in c). It would be better to set the split point at the nearest corner of the obstacle as in d), so area center path is far away enough from obstacle. The selected sector has dark blue color and the unselected one has a lighter color.

Correct split points localization is very important for a safe navigation. Using raw range measures the basic method usually provides a safe navigation, but there exists some situations, specially when doing goal tracking, where the position of the split point is not adequate yielding a dangerous situation. Sample situations depicted in figs 4 and 5 are taken from experiments described in [13]. The problem arise from selecting a sector, after split, with highly protruding corners. Selecting as split point the nearest segment end of those segments with some end between the area center and the bisector of original sector plus an offset, we can get sectors with less noticeable protuberances near the robot path, see fig. 4 d).

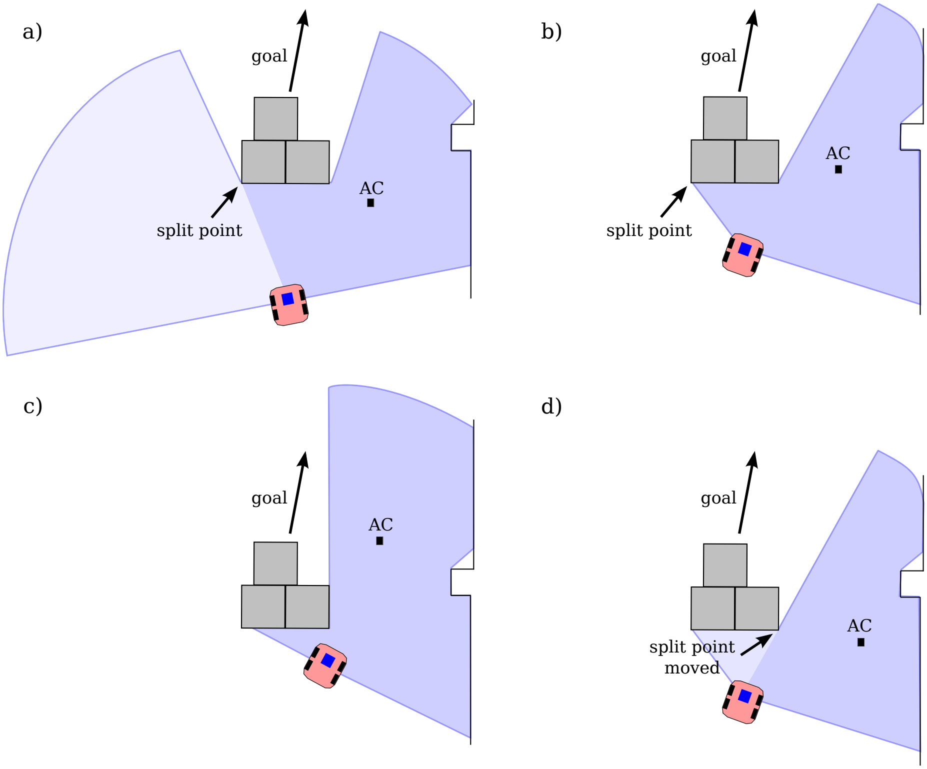


Fig. 5. Dynamic split points. In a) robot is slightly closer to the segment end marked with the arrow than to the other segment end. Then when area center becomes inaccessible inside the obstacle, the marked end is selected as split point and right sector is selected due to goal direction, this would lead to a dangerous situation b) and c), like in fig. 4, with the basic split point strategy. It would be better to change split point to the other end when the other segment end is closer to robot than split point, d), so area center path is far away enough from obstacle. The selected sector has dark blue color and the unselected one has a lighter color.

In order to get a safer and more robust navigation, specially in crowded or dynamic environments, we gave a step forwards enhancing split points behavior must be improved. Until new split points were static, but to response quickly to environment changes it would be better if they were dynamic. Even if the split point was well located at first time it is possible that a new protuberance appears in robot path without making area center inaccessible. In this case no split process is triggered, a basic solution for this problem is to change the position of a split point. So, if some segment end, located between split point and the bisector of current sector plus an offset, is closer to the robot, such segment end becomes the split point. This situation is depicted in fig. 5, and similar to one described in fig. 4, selected sector contains big protuberances. Changing split point to a new corner avoids the problem, see fig. 5 d).

Test performed in a robot simulator shows a better performance in robot navigation when using the improved method than the basic one. Robot exhibits an improved ability in turning corners in crowded environments and tracking goals, in a safe and efficient way. We reproduced the scenario of our real robot experiments from [13] in the simulator. When using the basic method we can observe many situations as those described previously in this section, a majority of these dangerous situations were avoided using the improved method.

7 Conclusions and Future Work

In this work we have presented a novel method for line extraction from range measures in polar coordinates. It has been designed for a noise model in which the measure errors have a standard deviation proportional to a known function of the expected measure, the rate constant may be unknown. This last fact allows our method to deal with different noise levels in the same scan. The method may be classified as a Line Regression with Clustering method and it is composed by three elements: a line regression method using EKF, a clustering procedure to look for adequate places in the scan that are useful to start EKF, and a merging procedure for similar adjacent segments.

The key for designing a filter that can deal with our noise model has been filter formulation. We have used and extended a filter formulation that is equivalent to regression models theory, so every result, property, test, etc., from regression theory are at our disposal. This fact provides tools for: line parameters estimation, noise variance estimation, estimators covariances matrix estimation, outliers detection, similar segments merging, etc., grounded in a well known and developed theory. As a future research, tools for scan matching in SLAM can be developed with the help of regression theory. Filter has linear time complexity.

We have used scale-space techniques for clustering by developing a filter for noise in order to get less fragmented features in the environment. Filter is based on a coarse characterization of the set of zero-crossing level curves generated by noise in the scale-space representation, followed by a statistical selection inside the set.

The clustering procedure usually provides adequate places, seeds, to start EKF estimation. Using outliers detection we can establish a segment end detection criterion. Possible overlap between segments are solved and similar segments are merged. Our C++ implementation provides execution times fast enough for use in robots.

We need to test extensively the whole process, at this moment only EKF has been tested extensively, and to make comparisons with other segmentation methods. Test in real world has been started using laser SICK-LMS 200 and Kinect device. Research on extending the method to curves and to 3D measures are at initial stages of development.

The first use for our segmentation method has been to improve the center of area method for navigation. By means of adequate split points selection, using segments ends, it is possible to get a more reliable and safer navigation.

References

1. Álvarez Sánchez, J.R., Mira Mira, J., de la Paz López, F., Cuadra Troncoso, J.M.: The centre of area method as a basic mechanism for representation and navigation. Robotics and Autonomous Systems 55(12), 860–869 (2007)
2. Álvarez-Sánchez, J.R., de la Paz Lépez, F., Troncoso, J.M.C., Sánchez, J.I.R.: Partial center of area method used for reactive autonomous robot navigation. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5602, pp. 408–418. Springer, Heidelberg (2009)
3. Borges, G.A.: A split-and-merge segmentation algorithm for line extraction in 2-d range images. In: Proceedings of the International Conference on Pattern Recognition, Washington, DC, USA, vol. 1, p. 1441. IEEE Computer Society Press, Los Alamitos (2000)
4. Borges, G.A., Aldon, M.-J.: Line extraction in 2d range images for mobile robotics. J. Intell. Robotics Syst. 40, 267–297 (2004)
5. Castellanos, J.A., Tardós, J.D. (eds.): Laser-based segmentation and localization for a mobile robot Robotics and manufacturing: Recent trends in research and applications, vol. 6. ASME Press, New York (1996)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 381–395 (1981)
7. Johnston, J., DiNardo's, J.: Econometric Methods, 4th edn. McGraw-Hill, Irwin (1996)
8. Lindeberg, T.: Scale-space for Discrete Signals. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 234–254 (1990)
9. Nguyen, V., Gächter, S., Martinelli, A., Tomatis, N., Siegwart, R.: A Comparison of line Extraction Algorithms using 2d Range Data for Indoor Mobile Robotics. Auton. Robots 23, 97–111 (2007)
10. Peña, D.: On internal robustification of Plackett-Kalman algorithm for recursive estimation of regression linear models. Trabajos de Estadística y de Investigación Operativa 36(1), 93–106 (1985) (in spanish)
11. Plackett, R.L.: Some theorems in least squares. Biometrika 37(1-2), 149–157 (1950)
12. Rosin, P.L.: Determining local natural scales of curves. Pattern Recognition Lett. 19, 63–75 (1994)
13. Álvarez Sánchez, J.R., de la Paz López, F., Cuadra Troncoso, J.M., de Santos Sierra, D.: Reactive navigation in real environments using partial center of area method. Robotics and Autonomous Systems 58(12), 1231–1237 (2010)
14. Siadat, A., Kaske, A., Klausmann, S., Dufaut, M., Husson, R.: An optimized segmentation method for a 2d laser-scanner applied to mobile robot navigation. In: Proceedings of the 3rd IFAC symposium on intelligent components and instruments for control applications (1997)
15. Siegel, A.F.: Robust regression using repeated medians. Biometrika 69(1), 242–244 (1982)
16. Taylor, R.M., Probert, P.J.: Range finding and feature extraction by segmentation of images for mobile robot navigation. In: Proceedings of IEEE International Conference on Robotics and Automation, april 1996, vol. 1, pp. 95–100 (1996)

17. Witkin, A.: Scale-space filtering: A new approach to multi-scale description. In: IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP 1884, vol. 9, pp. 150–153 (1984)
18. Zhang, S., Xie, L., Adams, M.D.: Feature extraction for outdoor mobile robot navigation based on a modified gauss-newton optimization approach. Robotics and Autonomous Systems 54(4), 277–287 (2006)

Analysis of EEG Mapping Images to Differentiate Mental Tasks in Brain-Computer Interfaces

Andrés Úbeda, Eduardo Iáñez, José M. Azorín, and Eduardo Fernández

Biomedical Neuroengineering Group, Miguel Hernández University of Elche
Av. de la Universidad S/N, 03202, Elche, Spain
{aubeda,eianez,jm.azorin,e.fernandez}@umh.es
http://nbio.umh.es

Abstract. This paper describes a study of a new classifier based on EEG mapping analysis in order to develop a Brain-Computer Interface (BCI) through computer vision techniques. To this end, the data from three different subjects (BCI Competition Data Set V) have been studied to show proper EEG maps of the three mental tasks registered. A new classifier based on image analysis of the EEG maps has been presented as a suitable way to distinguish between the different tasks, showing in which conditions of frequency and time the images obtained for each mental task can be best classified. The classifier has been tested obtaining the success percentage of classification of each subject showing that this kind of techniques are able to classify between three mental tasks with good results.

Keywords: Brain-Computer Interface (BCI), Electroencephalography (EEG), EEG Mapping, Computer Vision.

1 Introduction

A Brain-Computer Interface (BCI) is based on the processing of brain signals in order to generate commands to control external devices [1]. Brain signals can be registered using invasive or non-invasive methods. Invasive methods are based on the using of microelectrodes implanted directly in the brain. These techniques have been used in animals to determinate the movement intention [2] or to control a cursor in a screen [3]. In humans, the use of invasive techniques has ethical implications as well as medical risks. For this reason, electrodes are placed on the scalp of the patient to obtain the electroencephalographic (EEG) signals [4].

Non-invasive BCIs are commonly used in different applications such as the control of a robot [5,6] or the keyboard in a computer [7,8]. Non-invasive BCIs can be classified as evoked and spontaneous. In evoked interfaces, the EEG signals reflect an automatic response to external stimuli. This response is called evoked potential [9]. In spontaneous interfaces, the user performs a willful cognitive process or mental task that should be classified [7,10]. The registered EEG signals

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 246–255, 2011.

must be classified to differentiate between several cognitive processes or "mental tasks".

In order to distinguish between the different mental tasks, the EEG signals should be preprocessed, the most important features of the EEG signals are extracted, and finally these features are classified into the different classes to obtain the final mental task. The most common classifying of the processed signals is through different mathematical algorithms such as Linear Discriminant Analysis (LDA), Support Vector Machine (SVM) or Bayesian Classifiers, among others [11].

This paper studies a new approach in BCI classifiers through computer vision techniques. The main goal is to develop a classifier to differentiate between three different mental tasks. The features of the signals are extracted obtaining the power spectral density (PSD) and they are represented through images that show the frequency value in a period of time for each electrode (EEG mapping). These images are different for each mental state, as it is explained later, and it may be possible to classify them with a suitable image processing.

EEG mapping consists of plotting the electrical activity of the brain in a geometrical matrix. This approach gives a much more accurate and representative view of the mental activity obtained from the electrodes placed on the scalp. This can be done with a voltage/time representation or a frequency based representation. Nowadays, there are several works related to brain topography to differentiate several kinds of diagnoses, including some mental diseases whose origin is located in EEG alterations such as epilepsy [12,13] or schizophrenia [14]. EEG mapping has been also used in electrotherapy [15]. This kind of studies involve processing sessions of several minutes while on BCI interfaces the frequency of each decision is critic. This work shows the results obtained for a shorter period of processing time. To this end, the data are processed in windows of a few seconds to obtain the EEG maps. These maps are used by the classifier to distinguish between the three different mental tasks.

The remainder of this paper is organized as follows. In section 2, the steps made to create the different images used in this preliminary study are shown. Section 3 presents the classifier designed to differentiate the three mental tasks. In Section 4, the results obtained are shown. Finally, section 5 summarizes the main conclusions.

2 Image Obtention Protocol

The data set V of "mental imagery, multi-class" provided by IDIAP Research Institute for BCI Competition 2003 has been used to do the EEG mapping [16]. This data set contains data from 3 normal subjects during 4 non-feedback sessions (3 for training and 1 for test). The subjects made these experiments in 4 sessions on the same day, each one lasting 4 minutes and with 5-10 minutes breaks between them. For each session, the subjects performed three different tasks:

1. Imagination of repetitive self-paced left hand movements ("left" mental task).
2. Imagination of repetitive self-paced right hand movements ("right" mental task).
3. Generation of words beginning with the same random letter ("word" mental task).

The data are provided in two ways: raw EEG signals with a sampling rate of 512 Hz, and precomputed features. To obtain these features the raw EEG potentials were first spatially filtered with a surface Laplacian and then, every 62.5 ms (16 times per second), the power spectral density (PSD) in the band 8-30 Hz was estimated over the last second of data with a frequency resolution of 2 Hz.

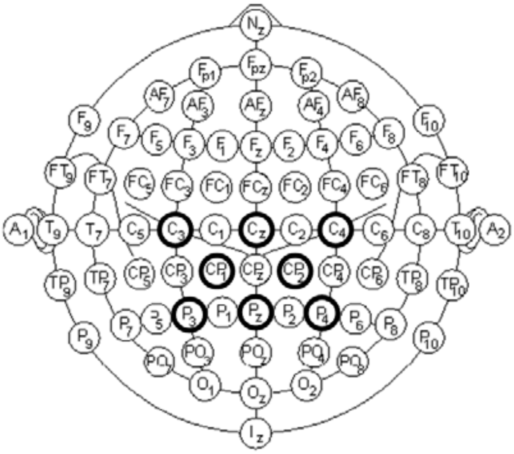


Fig. 1. 10/20 International System with the electrodes used marked in bold

Using the precomputed data, the EEG images have been obtained. The electrodes used to register the EEG signals are the 8 centro-parietal of the 10/20 International System: C3, Cz, C4, CP1, CP2, P3, Pz and P4 (see Figure 1). The final EEG sample is a 96-dimensional vector (8 channels with 12 frequency components).

The samples obtained have been plotted for each three users and sessions using Matlab. In Figure 2 an example of an EEG map can be seen. Each image represents an EEG map for a particular frequency and mental task. The axis show the position of the electrodes and the bar is scaled between 0 and 1 to improve the difference between each electrode. A total of 324 images (3 users x 3 sessions x 3 tasks x 12 frequencies) for a time interval of 5 seconds have been plotted. This amount of time has been obtained from a qualitative analysis of images from 5 seconds to 1 minute, averaging the data processed. This analysis showed that the minimum amount of time in which the different mental tasks were possible to differentiate was about 5 seconds.

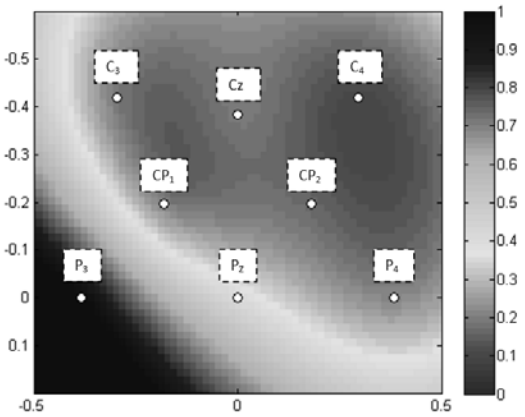


Fig. 2. Example of EEG map. The scale is normalized between 0 and 1 as it can be seen on the scale bar. Each electrode is placed in its particular position and the value generates the map.

3 Classification of Mental Tasks Using Computer Vision

After defining the protocol to obtain the EEG mapping images, a further image processing must be developed to obtain the mental tasks.

The technique proposed is based on a general image or pattern for each mental task and user that can be compared with each trial of data individually to determine which mental task is performing the user. The model of each task are obtained for each user through a training that consists of an offline session of several minutes. This data are processed in trials of 5 seconds that are averaged to obtain a final model for each mental task and used in the classifier.

First, the significant frequencies (the ones where the difference is substantial) have been chosen using a qualitative approach. Afterwards, a classification algorithm is proposed to differentiate between the different mental tasks.

3.1 Frequency Choice

As it was mentioned in Section 2, 12 different frequencies have been obtained after preprocessing the raw EEG signals. To classify the different mental tasks it is essential to have an appreciable difference between the maps obtained for each tasks. To that end, a qualitative analysis of the data of each subject has been made.

The most significant change between tasks appears in frequencies between 8-14 Hz. In particular, 12 Hz is the most significant frequency for Subject 1, 10 Hz for Subject 2 and, finally, 14 Hz for Subject 3. As it is shown in Figure 3, different shapes on the image can be clearly seen for left, right and word mental taks in the EEG mapping for these frequencies.

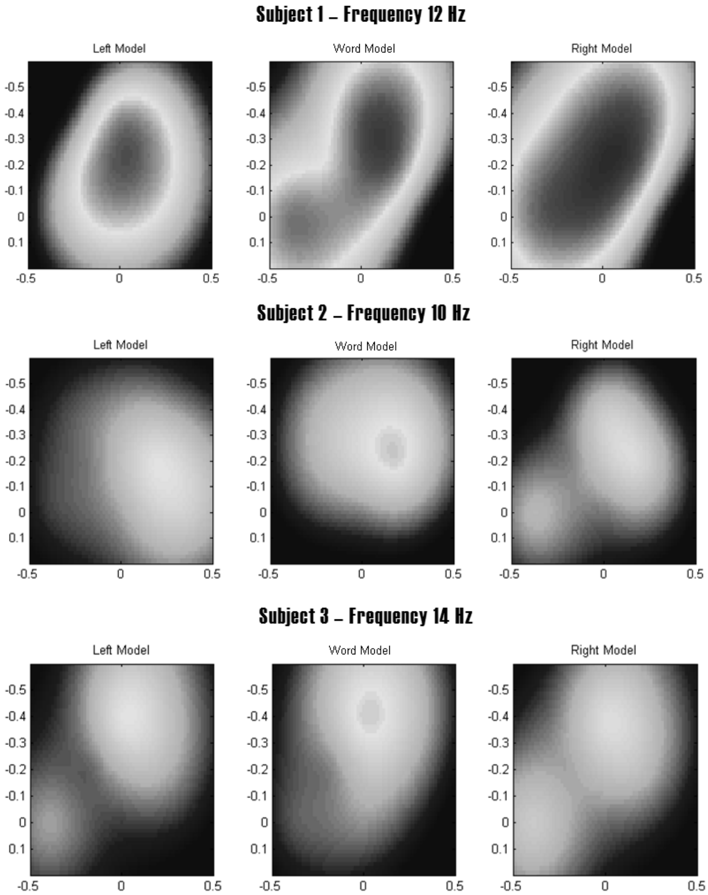


Fig. 3. Example of EEG maps obtained for each subject with the most significant frequencies used

3.2 Classifier

After deciding which are the most significant frequencies on each user, the EEG maps can be differentiated using an image comparison. A numerical index has been calculated to show the similarity between images. This value is obtained as the mean of the difference of all the points of the comparing images. As the original images that are subtracted have values between 0 and 1, the results obtained are the percentage of difference.

A classifier has been designed as it is shown in Figure 3 and is divided in two phases:

- **Models Obtention (Figure 4):** from the 4 sessions registered offline for each subject, 3 of them are selected to obtain the EEG maps models for each mental task (left, right and word). A processing window of 5 seconds is

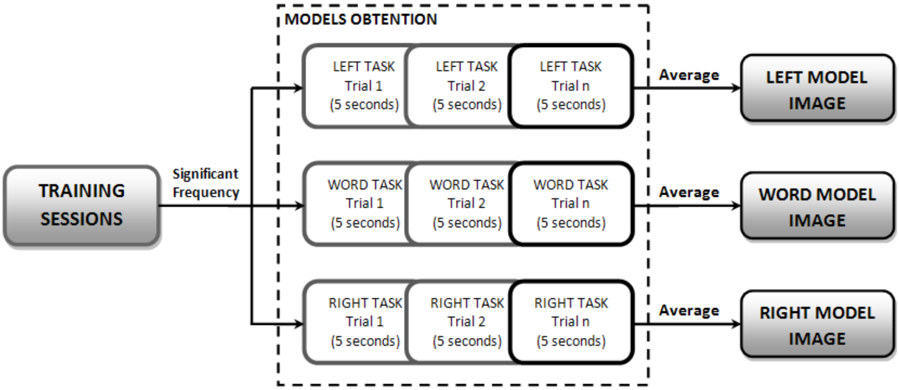

MODELS OBTENTION
TRAINING SESSIONS
Significant Frequency
LEFT TASK Trial 1 (5 seconds)
LEFT TASK Trial 2 (5 seconds)
LEFT TASK Trial n (5 seconds)
WORD TASK Trial 1 (5 seconds)
WORD TASK Trial 2 (5 seconds)
WORD TASK Trial n (5 seconds)
RIGHT TASK Trial 1 (5 seconds)
RIGHT TASK Trial 2 (5 seconds)
RIGHT TASK Trial n (5 seconds)
Average
Average
Average
LEFT MODEL IMAGE
WORD MODEL IMAGE
RIGHT MODEL IMAGE

Fig. 4. Models Obtention Protocol

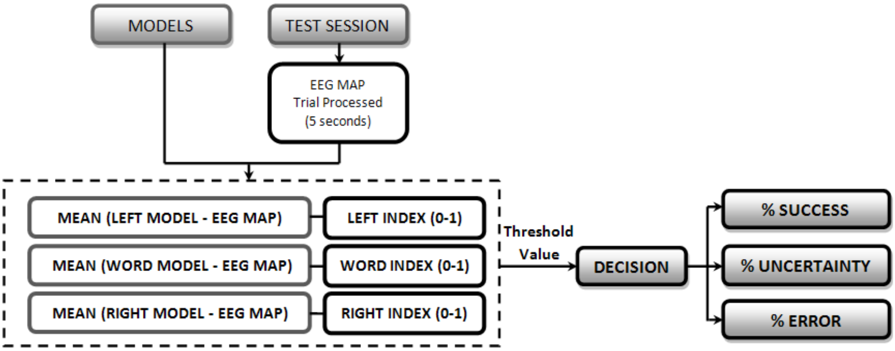


Fig. 5. Classification Algorithm

selected as well as the significant frequency previously studied. The data of each mental task are processed separately obtaining a collection of 5 second images that are averaged to get a unique EEG map for each task that will be used as the model to compare in the classifier.

- **Classification (Figure 5):** after obtaining the three models (left, right and word), the remaining set of data is used to test the classification. To that end, the data are processed in trials of 5 seconds obtaining the EEG map. This image is compared with the three models obtained before using the numerical index explained above. Afterwards, the success percentage is obtained using two modes:

  1. The one with greater similarity (left, right or word) is selected one obtaining directly the success percentage between the different tasks classified.
  2. A threshold is used to reject the trials that can not be clearly classified and introduces the concept of uncertainty in order to reduce the total error.

4 Results

The classifier has been tested with all three subjects using a cross validation. As it was mentioned, 3 sessions (75% of the data) have been used to obtain the models and the remaining session (25% of the data) has been used to test the classifier. The results obtained are shown in Table 1. Four combinations of sessions have been obtained showing the success percentage of the classifier. The success percentage is obtained as the average success of all three tasks. The success percentage of each task individually is shown for the best user (Subject 1) in Table 2.

Table 1. Offline results - Total success(%)

	Subject 1	Subject 2	Subject 3
123-4	68.6	62.8	48.6
124-3	64.6	59.2	44.8
134-2	72.0	60.5	50.4
234-1	51.5	56.4	34.0
Average	64.2	59.7	44.4

Table 2. Offline results for each mental task - Subject 1(%)

	Right	Word	Left
123-4	63.7	67.5	74.9
124-3	55.3	75.9	58.0
134-2	68.5	73.2	75.8
234-1	61.5	30.7	71.7
Average	62.3	61.8	70.1

The success percentages obtained prove that the EEG mapping method for BCI classification can be an effective new approach in BCI classifiers.

As it has been seen before, a variable of uncertainty has been introduced in a second group of experimental results. In this case, a threshold value has been used to reject trials that are not clearly similar to the models (0.8 for Subjects 1 and 3 and 0.75 for Subject 2). It can be verified that the error is reduced significantly compared to the previous results but affecting the final success percentage (Table 3). The percentages are obtained as the average percentage of all three tasks. Figure 6 shows the difference between the averaged results obtained without uncertainty (first column) and the averaged results obtained with uncertainty (second column) for each subject.

In Millán (2004) [17], similar results are obtained using a different classifier with the same data set. With this success percentage and trial time (5 seconds), the classifier is ready to work in a future online testing.

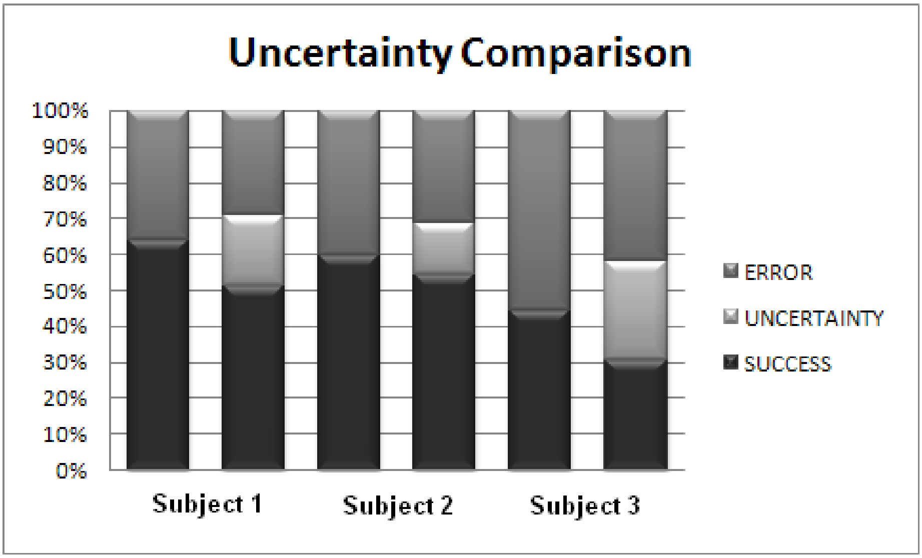


Fig. 6. Comparison between results including and not including uncertainty

Table 3. Offline results introducing uncertainty(%)

	Subject 1			Subject 2			Subject 3		
123-4	52.1	28.6	19.3	53.1	19.1	27.8	40.3	23.8	35.9
124-3	53.1	14.0	32.9	56.9	11.9	31.2	26.1	24.1	49.8
134-2	67.5	9.5	23.0	55.7	9.6	34.7	25.9	34.2	39.9
234-1	33.4	26.2	40.4	39.6	33.1	22.3	30.9	24.4	44.7
Average	51.5	19.6	28.9	51.3	18.5	30.2	30.8	26.6	42.6
	Success	Uncer.	Error	Success	Uncer.	Error	Success	Uncer.	Error

5 Conclusion

A computer vision technique has been proposed to classify mental tasks in a Brain-Computer Interface. This classifier uses EEG maps obtained from specific frequencies using data from three different subjects (BCI Competition Data Set V) and it is based on image comparison. The preliminary results show a good success percentage of classification so this new classifier can be useful to differentiate mental tasks in BCI. It is also expected that the results will improve using visual feedback.

As future works, online tests will be performed using visual feedback to analyze the usefulness of the classifier.

Acknowledgements

This work has been supported by the Ministerio de Ciencia e Innovacion of the Spanish Government through project DPI2008-06875-C03-03 and by the Conselleria d'Educació de la Generalitat Valenciana through grants BEST/2010/047 and FPA/2011/058.

References

1. Gao, X., Dignfeng, X., Cheng, M., Gao, S.: A BCI-based Environmental Controller for the Motion-Disabled. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11, 137–140 (2003)
2. Chapin, J.K., Moxon, K.A., Markowitz, R.S., Nicolelis, M.A.L.: Real-Time Control of a Robot Arm using Simultaneously Recorded Neurons in the Motor Cortex. Nature Neuroscience 2, 664–670 (1999)
3. Serruya, M.D., Harsopoulos, N.G., Paninski, L., Fellows, M.R., Donoghue, K.: Instant Neural Control of a Movement Signal. Nature 416, 141–142 (2002)
4. Millán, J.R., Ferrez, P.W., Buttfield, A.: Non Invasive Brain-Machine Interfaces - Final Report. IDIAP Research Institute - ESA (2005)
5. Inoue, S., Akiyama, Y., Izumi, Y., Nishijima, S.: The Development of BCI Using Alpha Waves for Controlling the Robot Arm. IEICE Transactions on Communications 91(7), 2125–2132 (2008)
6. Iáñez, E., Azorín, J.M., Úbeda, A., Ferrández, J.M., Fernández, E.: Mental Tasks-Based Brain–Robot Interface. Robotics and Autonomous Systems 58(12), 1238–1245 (2010)
7. Obermaier, B., Müller, G.R., Pfurtscheller, G.: Virtual Keyboard Controlled by Spontaneous EEG Activity. IEEE Transactions on Neural System Rehabilitation Engineering 11, 422–426 (2003)
8. Sirvent, J.L., Azorín, J.M., Iáñez, E., Úbeda, A., Fernández, E.: P300-based Brain-Computer Interface for Internet Browsing. In: IEEE International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS), pp. 615–622 (2010)
9. Bayliss, J.D.: Use of the Evoked Potential P3 Component for Control in a Virtual Environment. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11, 113–116 (2003)
10. Pfurtscheller, G., Neuper, C.: Motor Imagery and Direct Brain-Computer Communication. Proceedings of the IEEE 89, 1123–1134 (2001)
11. Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B.: A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces. Journal of Neural Engineering 4, 1–13 (2007)
12. Prasad, V.S., Murthy, J.M.K., Sailaja, S.: Surface Mapping of Spike Potential Fields: Visual vs. Quantitative EEG Analysis. Neurology India 50, 181–183 (2002)
13. Sebastián, M.V., Navascués, M.A., Valdizán, J.R.: Surface Laplacian and Fractal Brain Mapping. Journal of Computational and Applied Mathematics 189, 132–141 (2006)
14. Wong, M.T.H., Lieh-Mak, F.: Topographic Brain Mapping of EEG and Evoked Potentials in Chinese Normal and Psychiatric Patients - Preliminary Findings. J.H.K.C. Psych. 1, 6–11 (1991)

15. Kennerly, R.: QEEG analysis of cranial electrotherapy: A pilot study. Journal of Neurotherapy 8(2), 112–113 (2004)
16. BCI Competitions Web Source,
http://www.bbci.de/competition/?document=competition
17. Millán, J.d.R.: On the Need for On-line Learning in Brain-Computer Interfaces. International Joint Conference on Neural Networks (2004)

Design of a Hemispherical Antenna Array Receiver for Medical Applications

Mohammad Safar and Robert W. Newcomb

Department of Electrical & Computer Engineering, University of Maryland,
College Park, MD, USA
msafer10@hotmail.com, newcomb@eng.umd.edu

Abstract. A hemispherical antenna array consisting of 42 elements was proposed and simulated. The array is capable of receiving signals at 500 kHz. The array elements are distributed on a hemisphere to be placed on a human's head. The function of the array is to receive brain waves or signals from embedded electronics in the brain. By considering the problem as a boundary valued problem with the source signal or field being in the brain, the induced current on the antenna elements can be determined using boundary conditions. A beamforming technique for reception was used that made use of the orthogonality property of the *transverse electric* fields to enable the array to receive from one or multiple points from within the brain.

Index Terms: Hemispherical array, Boundary valued problem (BVP), Beaamforming.

1 Introduction

Antenna arrays have the ability to receive signals while improving the overall signal-to-noise since they allow for constructive interference in the desired point or direction of reception and destructive interference in all other directions. This property can be put in use when considering medical applications where there is a transmitter embedded inside the body and a receiver is needed to receive signals from that transmitter [1]-[4].

The reception is achieved by using some kind of a beamforming algorithm. This is usually done by first defining or calculating a real property of the signal that will be maximum in the direction where the signal is coming from and all what the designer have to do is try to scan for this maximum using some kind of an optimization scheme. According to [5], the algorithms for direction of arrival (DOA) estimation fall into the following 3 categories: (1) spectral-based algorithm; (2) subspace-based methods & (3) parametric methods. In the literature there are many kinds of reception strategies that fall in the above categories [6]-[13].

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 256–265, 2011.

In the case presented in this paper the array is a hemispherical array that is placed on the human head (or could be used for any other spherical surface) to receive signals from the brain either as brain waves or from embedded transmitters. A new beamforming technique is used that allows for reception from one or multiple directions.

2 Theoretical Analysis

2.1 Problem Statement

The design of the hemispherical array was done by solving an electromagnetic boundary valued problem in which the human head was modeled as a three layered hemisphere like the one shown in Fig. 1. Of course modeling the head as a perfect hemisphere is an assumption, but a necessary one to enable the use of the spherical coordinate system. The antenna elements are of course fixed onto a helmet which is in turn placed on the head but in the analysis it is assumed that the antennas are placed right above the head to reduce the number of layers and hence simplify the analysis. It is seen from the figure that there are four regions marked as (0), (1), (2) and (3). Region (0) is the outside of the head (i.e. free

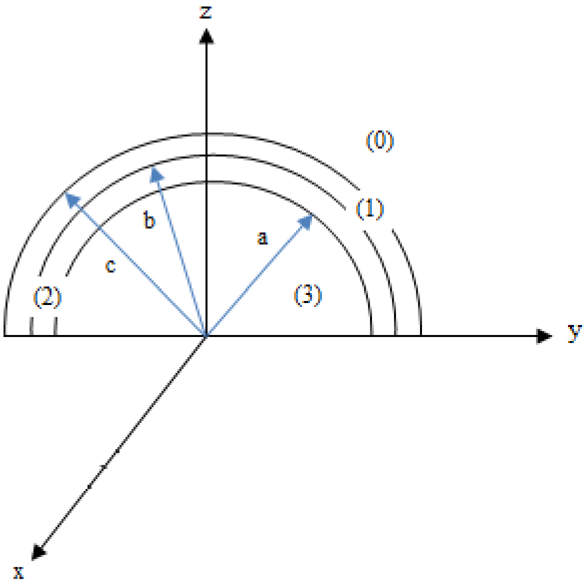


Fig. 1. A simple model of a human's head as seen from the front. There are four regions denoted by (0), (1), (2) and (3) which represent free space, the skull and brain respectively. The radius of the brain is a, while b denotes the radius from the center of the brain to the outer part of the skull and c for skin where the antenna elements are going to be placed.

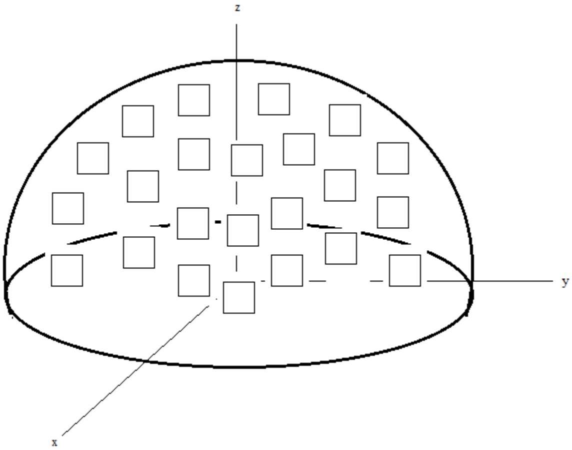


Fig. 2. Loop Antennas Placed on the Human head

space), region (1) is the skin, region (2) is the skull and finally region (3) is the brain. The 42 antennas are going to be placed on top of the skin surface and they will cover a hemispherical surface of radius c (Fig. 2).

Since the human head is approximated to be spherical, it is therefore obvious that the boundary value problem will be solved by utilizing the spherical coordinate system. According to [14, ch.6-1], any electromagnetic problem with spherical symmetry can be solved using the electric vector potential F_r and the magnetic vector potential A_r in spherical coordinates. Any electromagnetic wave can be decomposed into two orthogonal sets of functions which are the *transverse electric (TE)* F_r and the *transverse magnetic (TM)* A_r waves. Both vector potentials are solutions to the Helmholtz equation [14, ch.6-1] and can be used to find the fields in all four region. In the paper the fields will be expressed in terms of the TE modes only since the TM modes are smaller in comparison. The problem starts with the assumption that there is a localized (i.e. assumed as a delta function) radiating magnetic field in region (3) at a distance r_s coming from the direction *(θ_s,φ_s)* (the subscript s denotes source). Since the H-field is localized it can be described by the following equation which.

$$\vec{H}_s = \hat{\theta} H_o \delta(\theta-\theta_s)\delta(\varphi-\varphi_s) + \hat{\varphi} H_o \delta(\theta-\theta_s)\delta(\varphi-\varphi_s) \tag{1}$$

Where H_o represents the magnitude of the H-field. To solve the BVP equation (1) must be expanded in terms of the TE modes as shown in (2):

$$\vec{H}_s = -i\sqrt{\frac{\varepsilon_3}{\mu_0}} \sum_{l=1}^{\infty} \sum_{m=-l}^{l} F_{lm} \nabla \times \left[j_l(k_3 r)\vec{r} \times \nabla Y_{lm}(\theta,\varphi) \right] |_{r=r_s} \tag{2}$$

To find the coefficient F_{lm} of the TE expansion (1) and (2) are equated and then the orthogonality property of the TE modes is applied. F_{lm} is given by the following equation.

$$F_{lm} = \frac{k_3 r_s \sqrt{\mu_o} \,/\, J_l'(k_3 r_s)}{i\sqrt{\varepsilon_3}\, l(l+1)} \left[\sin\theta_s \frac{\partial Y^*_{lm}(\theta_s,\varphi_s)}{\partial\theta} - im Y^*_{lm}(\theta_s,\varphi_s) \right] \tag{3}$$

Where $J(k_3 r) = k_3 r j_l(k_3 r)$ and $j_l(k_3 r)$ [14, ch.6-1] is the spherical Bessel function, J_l' is the derivative w.r.t r, $i = \sqrt{-1}$, μ_0 is the permeability of free space, ϵ_3 is the relative permittivity of region (3) and $Y_{lm}(\theta, \varphi)$ is the spherical harmonic with l and m as the indices of the expansion. The wavenumber k_3 is given by:

$$k_3 = \omega\sqrt{\mu_0\varepsilon_0(\varepsilon_3 + \frac{i\sigma_3}{\omega\varepsilon_0})} \tag{4}$$

Where ε_0 is the permittivity of free space, $\omega = 2\pi f$ is the frequency in radian/sec (f=*500kHz*) and σ_3 is the conductivity of region (3).

Beginning with the field in region (3) as the source, one can use the boundary conditions presented in (5)-(10) to solve for the current density of each antenna element given by (8).

$$\hat{r}\times(\vec{E}_1 - \vec{E}_0) = 0 \quad \text{for } r = c \tag{5}$$

$$\hat{r}\times(\vec{E}_2 - \vec{E}_1) = 0 \quad \text{for } r = b \tag{6}$$

$$\hat{r}\times(\vec{E}_3 - \vec{E}_2) = 0 \quad \text{for } r = a \tag{7}$$

$$\hat{r}\times(\vec{H}_1 - \vec{H}_0) = \vec{J}_i(\theta,\varphi) \quad \text{for } r = c \tag{8}$$

$$\hat{r}\times(\vec{H}_2 - \vec{H}_1) = 0 \quad \text{for } r = b \tag{9}$$

$$\hat{r}\times(\vec{H}_3 - \vec{H}_2) = 0 \quad \text{for } r = a \tag{10}$$

Where $\boldsymbol{H}_j$ and $\boldsymbol{E_j}$ represent the magnetic intensity (H-field) and electric field in the region (j) and $\boldsymbol{J_i}$ is the current density of the i-th antenna element.

By solving the BVP by using equation (2) and (5)-(10), one can derive the expression of the current density induced at the ith antenna. After determining the current density it is very easy to determine the current signal induced at the ith antenna and this is given by (11).

$$I_i = L\sum_{l=1}^{\infty}\sum_{m=-l}^{l} F_{lm}\Psi_l \frac{im}{\sin\theta_i} Y_{lm}(\theta_i,\varphi_i) \tag{11}$$

Where:

$$\begin{aligned}\Psi_l = & \frac{\eta_0}{k_0 c}\frac{H_l^{'(2)}(k_0 c)}{H_l^{(2)}(k_0 c)}\left\{H_l^{(1)}(k_1 c)\left[\frac{\tau_l J_l(k_3 a)}{\gamma_l H_l^{(1)}(k_2 a)+H_l^{(2)}(k_2 a)}\right]+\right.\\ & \left. H_l^{(2)}(k_1 c)\left[\frac{J_l(k_3 a)}{H_l^{(2)}(k_1 b)}\left[\frac{\gamma_l H_l^{(1)}(k_2 b)+H_l^{(2)}(k_2 b)-\tau_l H_l^{(1)}(k_1 b)}{\gamma_l H_l^{(1)}(k_2 a)+H_l^{(2)}(k_2 a)}\right]\right]\right\}\\ & -\frac{\eta_1}{k_1 c}\frac{\tau_l J_l(k_3 a)H_l^{'(1)}(k_1 c)}{\gamma_l H_l^{(1)}(k_2 a)+H_l^{(2)}(k_2 a)}\\ & -\frac{\eta_1}{k_1 c}\frac{J_l(k_3 a)H_l^{'(2)}(k_1 c)}{H_l^{(2)}(k_1 b)}\left[\frac{\gamma_l H_l^{(1)}(k_2 b)+H_l^{(2)}(k_2 b)-\tau_l H_l^{(1)}(k_1 b)}{\gamma_l H_l^{(1)}(k_2 a)+H_l^{(2)}(k_2 a)}\right]\end{aligned} \tag{12}$$

$$\gamma_l = \frac{H_l^{'(2)}(k_2 a)J_l(k_3 a)-H_l^{(2)}(k_2 a)J_l^{'}(k_3 a)}{H_l^{(1)}(k_2 a)J_l^{'}(k_3 a)-H_l^{'(1)}(k_2 a)J_l(k_3 a)}$$

$$\tau_l = \frac{\left\{\begin{array}{l}\left[\gamma_l H_l^{(1)}(k_2 b)+H_l^{(2)}(k_2 b)\right]H_l^{'(2)}(k_1 b)\\ -\left[\gamma_l H_l^{'(1)}(k_2 b)+H_l^{'(2)}(k_2 b)\right]H_l^{(2)}(k_1 b)\end{array}\right\}}{\left[H_l^{(1)}(k_1 b)H_l^{'(2)}(k_1 b)-H_l^{'(1)}(k_1 b)H_l^{(2)}(k_1 b)\right]} \tag{13}$$

Where L is the dimension of the antenna element, $H_l^{(1,2)}(k,r) = krh_l^{(1,2)}(k,r)$ and $h_l^{(1,2)}(k,r)$ [14, ch.6-1] is the spherical Hankel function of the first and second kind and $H_l^{'}$ is the derivative w.r.t r. The coefficients presented in (12) and (13) are simply the result from solving the BVP.

2.2 Beamforming Technique

After obtaining the received current signal equation in the previous section it is now possible to consider the beamforming technique that will allow the array to efficiently scan the environment. By proper scanning the array will determine the source from which the source H-field originated.

The idea behind the technique is based on the orthogonality principle of the TE modes. To begin the analysis the received input current at the ith antenna element is expressed as follows.

$$I_{it} = I_i + n_i \tag{14}$$

Where I_i is given by (11) and n_i is a zero mean white Gaussian noise. Each signal I_i will be multiplied by a weight factor (i.e. magnitude and phase) g_i and then all the signals from all the elements are added up to produce one output current I_{out} given by equation (15).

$$I_{out} = \sum_{i=1}^{N} I_{it} g_i = \sum_{i=1}^{N}\left(I_i g_i + n_i g_i\right) = FSg^V + Ng^V \tag{15}$$

The first part of (15) gives the output current in a summation form while the second part expresses the current in a matrix form.

Where:

$$F=\left[F_{1,-1}(\theta_s,\varphi_s)\quad F_{1,0}(\theta_s,\varphi_s)\quad F_{1,1}(\theta_s,\varphi_s)\quad .\quad .\quad F_{M,M}(\theta_s,\varphi_s)\right] \tag{16}$$

$$g^V=\begin{bmatrix} g_1\\ g_2\\ g_3\\ .\\ .\\ g_N \end{bmatrix} \tag{17}$$

$$S=\begin{bmatrix} \Psi_1\frac{-j}{\sin\theta_1}Y_{1,-1}(\theta_1,\varphi_1) & \Psi_1\frac{-j}{\sin\theta_2}Y_{1,-1}(\theta_2,\varphi_2) & . & . & . & \Psi_1\frac{-j}{\sin\theta_N}Y_{1,-1}(\theta_N,\varphi_N)\\ 0 & 0 & . & . & . & 0\\ \Psi_1\frac{j}{\sin\theta_1}Y_{1,1}(\theta_1,\varphi_1) & \Psi_1\frac{j}{\sin\theta_2}Y_{1,1}(\theta_2,\varphi_2) & . & . & . & \Psi_1\frac{j}{\sin\theta_N}Y_{1,1}(\theta_N,\varphi_N)\\ . & . & . & & & .\\ . & . & & . & & .\\ \Psi_M\frac{jM}{\sin\theta_1}Y_{M,M}(\theta_1,\varphi_1) & \Psi_M\frac{jM}{\sin\theta_2}Y_{M,M}(\theta_2,\varphi_2) & . & . & . & \Psi_M\frac{jM}{\sin\theta_N}Y_{M,M}(\theta_N,\varphi_N) \end{bmatrix} \tag{18}$$

$$N=\left[n_1\quad n_2\quad n_3\quad .\quad .\quad n_N\right] \tag{19}$$

F is a vector that contains the information about the source signal, S is a matrix that contains the terms of the double sum present in (11), g^V is a column vector that contains the adjustable weights and N is a row vector that contains the noise for each antenna element.

The aim in this problem is to use the vector g^V to transfer the matrix S into the following desired vector.

$$S_{Desired}=\begin{bmatrix} \frac{-j}{\sin\theta}Y_{1,-1}(\theta,\varphi)\\ 0\\ \frac{j}{\sin\theta}Y_{1,1}(\theta,\varphi)\\ .\\ .\\ \frac{jM}{\sin\theta}Y_{M,M}(\theta,\varphi) \end{bmatrix} \tag{20}$$

Reducing the matrix from S to the vector $S_{Desired}$ allows for the use of the orthogonality principle. When considering the product between F and $S_{Desired}$ the output current becomes:

$$
\begin{aligned}
I_{out} &= FS_{Desired} + Ng^{V} = \sum_{l=1}^{M=7}\sum_{m=-l}^{l} F_{lm}\frac{jm}{\sin\theta}Y_{lm}(\theta,\varphi) + \sum_{i=1}^{N} n_i g_i \\
I_{out} &\approx \delta(\theta-\theta_s)\delta(\varphi-\varphi_s) + \sum_{i=1}^{N} n_i g_i
\end{aligned}
\tag{21}
$$

It is clear from (21) that the resulting current is desired to be a delta-like form in the control angles (θ, φ) with some added noise. The scanning process is done in the sense that the control angles (θ, φ) are going to be varied from 0 to π for θ and from 0 to 2π for φ. For each control angle set (θ, φ), I_{out} is calculated by first finding the optimal set of weight factors g^V. When the calculation is done for all the angle sets, plotting $I_{out}(\theta, \varphi)$ will show that the output will have its maximum value at (θ_s, φ_s) which is the aim of the array. In this way by using some kind of a maximum seeking circuit and after that determining at which control angle set this maximum was found, the direction of arrival detection is possible.

To transform S to $S_{Desired}$ the following problem must be solved for each control angle set.

$$
\arg\min_{g} \left\| Sg^{V} - S_{Desired} \right\|^{2} \tag{22}
$$

The above minimization statement can be easily solved for with respect to the vector g^V by setting the vector to:

$$
g_{opt}^{V} = (S^{H}S)^{-1}S^{H}S_{Desired} \tag{23}
$$

Note that the same analysis applies if more than one source is being detected.

3 Matlab Simulation Results

The theory presented in Section II is simulated in MATLAB. As mentioned earlier 42 elements were used in the simulation. Each element has a dimension of *L=2cm* and the frequency of reception is taken to be 500 kHz. The spacing between the centers of the elements is $\approx 4\,\text{cm}$. For an operating frequency of 500 kHz the electromagnetic properties of the human head in region 0-3 are as follows (as given in [15]): region (0) has the free space permittivity and permeability; region (1) has $\varepsilon_1 = 38.87\epsilon_0$, $\mu_1 = \mu_0$ & $\sigma_1 = 1.88\,\text{S/m}$; region (2) has $\varepsilon_2 = 19.34\epsilon_0$, $\mu_2 = \mu_0$ & $\sigma_2 = 0.59\,\text{S/m}$; and region (3) has $\varepsilon_3 = 51.8\epsilon_0$, $\mu_3 = \mu_0$ & $\sigma_3 = 1.5\,\text{S/m}$. The simulations in Figures 3-5 show how the output current I_{out} in (21) varies with the angular parameters θ and φ. Note that all the simulations made are for a radial distance of $r_{(0)} = 5\,\text{cm}$ from the center of the coordinate system as shown in Fig. 1.

It is clear from the figures that the beamforming technique is successful in determining the direction of the source H-field (θ, φ) indicated by each figure. The origin of the source clearly corresponds to the maximum value in the output current at every case. For the case of Fig. 5b it is clear that the technique is

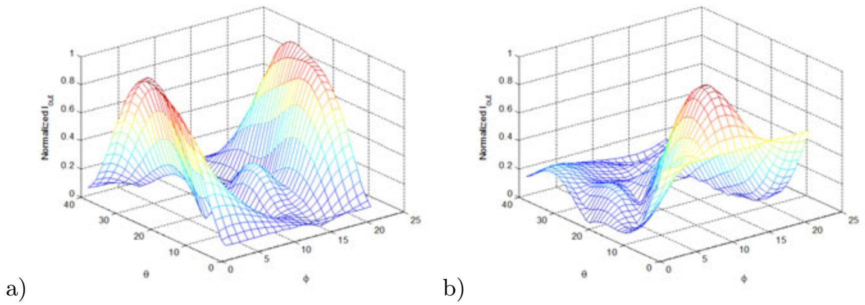


Fig. 3. *a)* $I_{out}(\theta, \varphi)$ for $\theta_s = \frac{\pi}{2}$ & $\varphi_s = 0$. *b)*$I_{out}(\theta, \varphi)$ for $\theta_s = \frac{\pi}{4}$ & $\varphi_s = \pi$

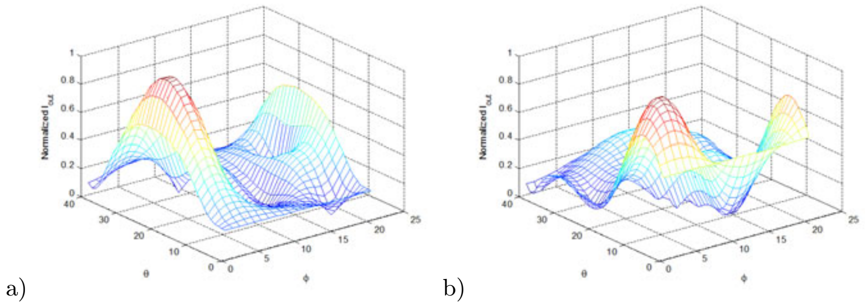


Fig. 4. *a)* $I_{out}(\theta, \varphi)$ for $\theta_s = \frac{\pi}{4}$ & $\varphi_s = \frac{\pi}{2}$. *b)*$I_{out}(\theta, \varphi)$ for $\theta_s = \frac{\pi}{2}$ & $\varphi_s = \frac{\pi}{4}$

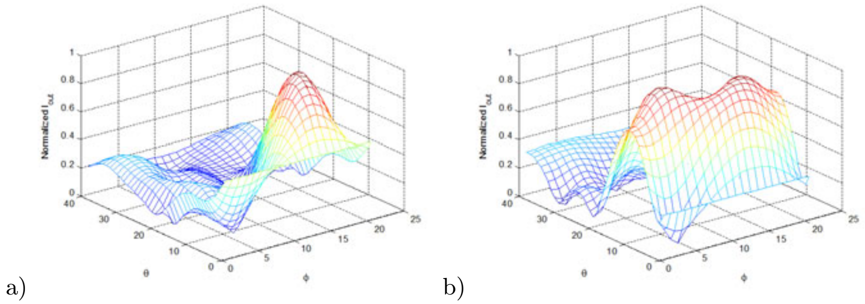


Fig. 5. *a)* $I_{out}(\theta, \varphi)$ for $\theta_s = \frac{\pi}{4}$ & $\varphi_s = \frac{3\pi}{2}$. *b)*$I_{out}(\theta, \varphi)$ for $\theta_{s1} = \frac{\pi}{4}$ & $\varphi_{s1} = \frac{3\pi}{2}$ *and* $\theta_{s2} = \frac{\pi}{4}$ & $\varphi_{s2} = \frac{\pi}{2}$

applicable for multiple source detection (in this example two sources). It is clear from Fig. 5b that there are two maximum values for the current which correspond to the two location of the two source H-fields. Another remark is the scaling used in the figures. The simulation was carried out for values of θ ranging from 0 to π and φ ranging from 0 to 2π. In the plots the θ range is divided into 40 points while the φ range is divided into 20 points. For example, for the φ-axis the point 10 corresponds to $\varphi = 10 * \frac{2\pi}{20} = \frac{\pi}{2}$ and for the θ axis the same point corresponds to $\theta = 10 * \frac{\pi}{40} = \frac{\pi}{4}$.

4 Conclusions and Future Work

A theoretical hemispherical antenna array is designed and simulated to receive brain waves or signals embedded in the human head. A new beamforming technique that makes use of the orthogonality property of the TE modes is devised and used. The beamforming technique used is successful in receiving from single or multiple sources. The array can be applied to any other application where spherical geometry is involved where the task is to receive signals from within this spherical geometry. The maximum seeking circuit for the output current is under investigation and also the antenna elements and all the radio-frequency RF front-end circuitry is being designed for simulation.

References

[1] Want, R.: An Introduction to RFID Technology. In: IEEE CS and IEEE ComSoc. (2006)

[2] Rueangsi, N., Thanachayanont, A.: Coil Design for Opt imum Operating Range of Magnetically-Coupled RFID System. In: IEEE ISCIT, pp. 1199–1202 (2006)

[3] Atluri, S., Ghovanloo, M.: Incorporating Back T elemet ry in a Full-Wave CMOS Rect ifier for RFID and Biomedical Applicat ions. IEEE, Los Alamitos (2007)

[4] Fotopoulou, K., Flynn, B.: Optimum Antenna Coil Structure for Induct ive Powering of Passive RFID Tags. In: IEEE International Conference in RFID, pp. 71–77 (2007)

[5] Amin, M., Rahman, A., Bin-Mahbub, S., Ahmed, K., Chowdhury, Z.: Estimat ion of Direct ion of Arrival (DOA) Using Real-T ime Array Signal Processing. In: 5th International Conference on Electrical and Computer Engineering ICECE 2008, Dhaka, Bangladesh, December 20-22, pp. 422–427 (2008)

[6] Vasylyshyn, V.I.: Antenna Array Signal Processing with High -Resolution By Modified Beamspace Music Algorithm. In: International Conference on Antenna Theory and Techniques, Sevastopol, Ukraine, September 17-21, pp. 455–457 (2007)

[7] Zooghby, A., Christodoulou, C.C., Georgiopoulos, M.: A Neural Network-Based Smart Antenna for Multiple Source Tracking. IEEE Transaction on Antennas and Propagation 48(5), 768–776 (2000)

[8] Sarevska, M., Milovanovic, B.: Alternative Signal Detection For Neural Network-Based Smart Antenna. In: 7th Seminar on Neural Network Applications in Electrical Engineering, Neural-2004, September 23–25, pp. 85–89 (2004)

[9] Sarevska, M., Salem, A.: Antenna Array Beamforming using Neural Network. World Academy of Science, Engineering and Technology 24, 115–119 (2006)

[10] Caylar, S., Dural, G., Leblebicioglu, K.: Neural Network Method for Direction of Arrival Estimation with Uniform Cylindrical Microstrip Patch Array. IET Microwaves, Antennas &Propagation (2008)

[11] Svantesson, T., Wennstrom, M.: High Resolution Direction Finding Using a Switched Parasitic Antenna, http://www.signal.uu.se/Publications/pdf/c0111.pdf

[12] Chen, H.H., Chan, S.C., Zhang, Z.G., Ho, K.L.: Adaptive Beamforming and Recursive DOA Estimation Using Frequency-Invariant Uniform Concentric Spherical Spheres. IEEE Transactions on Circuit and Systems 55(10), 3077–3089 (2008)

[13] Allen, B., Ghavami, M.: Adaptive Array Systems. Wiley, Chichester (2005)

[14] Harrington, R.F.: Time-Harmonic Electromagnetic Fields. In: IEEE Press Series on Electromagnetic Wave Theory, ch. 6, pp. 264–269 (2001)

[15] Marinova, I., Mateev, V.: Electromagnetic Field Modeling in Human Tissue. World Academy of Science, Engineering and Technology 64, 298–303 (2010)

[16] Partal, P., Mautz, R., Arvas, E.: Radiation from a Circular Loop in the Presence of a Spherically Symmetric Conducting or Dielectric Objects. IEEE Trans. Antenna and Propagation 48(10), 1646–1652 (2000)

[17] Li, Z., Duraiswami, R.: Flexible and Optimal Design of Spherical Microphone Arrays for Beamforming. IEEE Trans. Speech & Audio Processing 15(2), 702–714 (2007)

Long Term Modulation and Control of Neuronal Firing in Excitable Tissue Using Optogenetics

L. Humphreys[1,3], J.M. Ferrández[1,2], and E. Fernández[1,3]

[1] Instituto de Bioingeniería, Universidad Miguel Hernández, Alicante
[2] Dpto. Electrónica, Tecnología de Computadoras, Univ. Politécnica de Cartagena
[3] CIBER-BBN
lhumphreys@umh.es

Abstract. Since the initial demonstration of nerve excitation and the subsequent action potential generation by Hodgkin and Huxley in 1952, most efforts in modulating or restoring neural activity to cure diseases or injury have concentrated on using neural interfaces for electrical stimulation with electrodes. However, it was soon appreciated that repeated chronic stimulations necessary for lasting rehabilitation could have its drawbacks. Namely, the eventual degradation of tissue and electrodes, issues of biocompatibility and immune responses to foreign objects. Nevertheless, new innovative methods are emerging which can improve the quality and duration of neural stimulations. Here we review and suggest an alternative approach to modulate activity using optogenetics in therapy.

1 Introduction

Neuroprosthetic devices have been implemented in therapies based on their ability to interface with the central and peripheral nervous system. These devices normally consist of electrodes which are chronically implanted to perform stimulation and recordings of excitable cells, effectively monitoring and re-introducing the activity absent to the effected areas. Therapeutic applications are widespread ranging from, limb prostheses, retinal and cortical prostheses, spinal cord innervation, stroke, pain and depression alleviation to name a few [1] [2]. Some devices have had profound implications on the quality of patient's lifes, in particular, cochlear implants [3] and deep brain stimulation (DBS) for tremor treatment in Parkinson's disease [4]. However, chronic stimulation of excitable tissue has also been met with considerable complications which eventuate in the deterioration of stimulation and recording quality. Most efforts have focused on resolving issues of biocompatibility which primarily depend on the mechanical properties of the neural interface and interaction of the surrounding tissue. Other examples include micromovements of the electrodes, trauma from the initial insertion of which can lead to intrinsic foreign body responses and glial inflammation and neuronal loss near the implant [5]. Normally, electrical stimulation induces a functional response by depolarizing the membranes of excitable cells through the application of biphasic current pulses. The importance of a charge balanced

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 266–273, 2011.

waveform is to avoid damage to the electrodes and surrounding tissue. However, even if the stimulus pulse is charge balanced an electrode may still be polarized during delivery resulting in the subsequent damage of the electrode and tissue [6]. Similarly an unwanted effect can arise through the electrolysis of surrounding water with subsequent changes to pH and gas formation [7] [8]. Thus alternative methods of cellular activation which avoid these issues are essential if chronic stimulations are to be used in therapies. In this work we propose the use of the light activated cation channel, Channelrhodopsin-2 (ChR2) as an alternative means to modulate and control neural activity. Long term photoactivation of ChR2 expressing hippocampal cultures [9] and slices [10] has already been demonstrated using LEDs and we propose this as a possible alternative method of chronically activating neural tissue for therapeutic purposes circumventing some of the issues faced by electrical stimulation.

2 Channelrhodopsin-2

Extracellular electrodes have limitations in their spatial resolution and intracellular electrodes rely on mechanical stability, are invasive and ultimately lead to cell death. However, the use of light activated channels permits to manipulate neural activity within targeted cells with high temporal precision and in a non-invasive manner [11]. Moreover, the use of light for chronic stimulation can also be projected onto tissue with great spatial accuracy and can be used to stimulate specific neuronal regions such as the axons or dendrites [12]. The first attempts to convey photoexcitabilty in non photoresponsive neural cells was accomplished by taking advantage of the invertebrate rather than the vertebrate phototransduction system as it differs in the polarity of current it produces. In invertebrates light mobilizes a class of G protein (Gq/11) which activates phospholipase C (PLC). This results in non-specific cation channels in the plasma membrane opening which in turn depolarize the cell [13]. The potential of this invertebrate light cascade to depolarise other cells was recognized and it was subsequently determined that the minimum protein requirement for conveying phototransduction was the G protein couple receptor NinaE, a G protein aq subunit and arrestin, otherwise referred to as chARGe [14]. However, even with this cohort of proteins present, light responses are slow and small in magnitude and require the exogenous expression of three different genes. Since then, two distant relatives of rhodopsin, Channelrhodopsin 1 and 2, were identified in the unicellular green algae *Chlamydomonas reinhadrtii.* Both these channels are gated by visible light and are thought to enable phototaxis by coupling to specific transducers. ChR1 has been shown to be selective towards protons [15]. ChR2 differs in that it is a rapidly gated light-sensitive non-selective cation channel with conductance for $H^+ > Na^+ > K^+ > Ca^{2+}$. Within the algae the photoreceptors normally are located in the cytoplasmic membrane and situated in the eyespot region [16]. Amino acids 1-315 of ChR2 have been established to convey photoresponsiveness [17]. As photocurrents are produced within $<$ 50ms of a light flash in the algae it was understood that the photoreceptor and the channel form a complex or are one and the same. ChR2 is activated with blue light (480nm) in which the all-trans form of retinal isomerizes into its cis configuration opening the channel (Fig.1). Under normal conditions the inward flood of cations depolarizes the cell resulting in the generation of action potentials. The channel can return to the all-trans ground state without the necessity for other enzymes, making it ideal for stimulating neurons with millisecond precision. Recovery from inactivation is also dependent on extracellular pH with it being slower in high pH and faster in low pH. Pulses of H^+ given during the dark periods accelerates the recovery of the channel from desensitization suggesting that it is dependent also on the protonation of certain amino acids situated on the extracellular side. It has been suggested that the amino acid Glu-123 is the main culprit as it is predicted to be exposed on the extracellular side and mutating it to Gln abolished peak photocurrents [17].

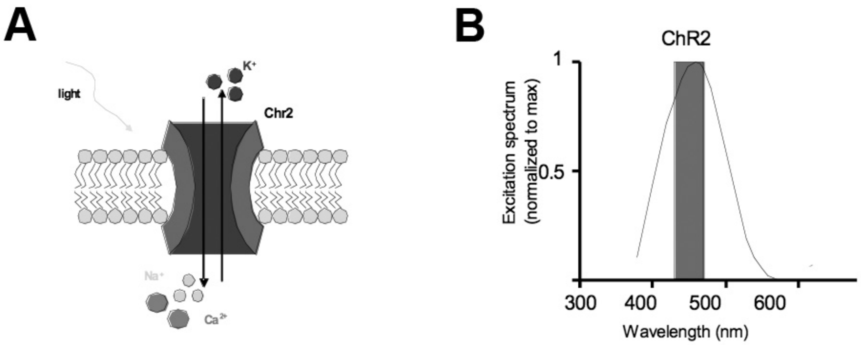


Fig. 1. Photostimulation of Channelrhodopsin-2. A: Illumination of ChR2 with blue light (bandwidth 450-500nm) activates the channel allowing cations to pass through resulting in depolarization of excitable cells. B: Action spectra of ChR2 indicating maximum activation between 470 and 490nm in the blue spectrum.

Since its discovery, ChR2 has been successfully expressed in mammalian cells [17] [18], Drosophila [19], *C. Elegans* [20] as well as in vivo in rodent inner retinal cells [21], motor cortex [22], spinal chord and hindbrain [23] to name a few. Other advantages of this tool are also the apparent lack of toxicity to cells that have been genetically targeted with ChR2 [22]. Importantly; no significant differences have been reported between the electrical properties of ChR2 +ive and -ive cells confirming that the expression of protein does not affect the resting properties and action potentials generation in transgenic mice [24]. As ChR2 is encoded by a single gene less than 1kb long it can be targeted with relative ease to distinct cell populations through genetic techniques. ChR2 has been tagged to various fluorophore markers through molecular engineering techniques making it possible to visually identify neurons expressing the channel. One slight disadvantage is that visualization of ChR2+ive neurons with GFP-YFP is not possible without concurrently activating the channel.

Importantly the presence of retinal is crucial for ChR2 to function properly. When expressed in the mammalian brain exogenous delivery of its cofactor is not required due to the presence of endogenous retinoids. In cultures this problem is overcome by the presence of B-27 in the culture medium which contains retinyl acetate. All these properties of ChR2 make it an ideal candidate for use in long term in vivo stimulation (Fig 2).

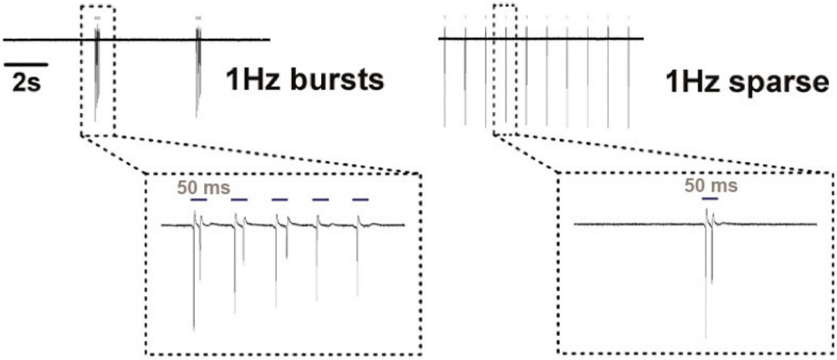


Fig. 2. Example on-cell configuration recordings from a ChR2+ve neuron during 1Hz burst (left) and 1Hz sparse (right) LED photostimulation. Modified from Grubb and Burrone 2010 [9].

3 From Probe to Therapeutic Tool

For ChR2 to make the transition from probe to therapeutic tool it will have to meet certain requirements. Firstly, it must not be toxic or interefere with normal molecular mechanisms. Most papers in which ChR2 has been expressed in exciteable cells have not reported any adverse effects [22] [24]. However, one possible complication that might arise from chronic photostimulations necessary for lasting rehabilitation which must be addressed is phototoxicity in response to constant light exposure. There are reports that blue light induces mitochondrial DNA damage through the production of free radicals in epithelial cells after 3 hours of light exposure [25]. Long term tests will have to be carried out to ensure that in vivo stimulations can withstand such pressures through naturally occuring antioxidants. Secondly it needs to be stably expressed and functional in the order of years to decades. So far ChR2 has been expressed in vivo in the primate cortex and hippocampus for several weeks [26] and activated for more than 2 days in hippocampal cultures [9] and 24hrs in organotypic slices [10]. Thirdly a suitable mechanism of ChR2 delivery must be selected. Primate-appropriate viral vectors with human promoters have already been tested with success [26]. Forthly, appropriate means of light delivery must be selected. This will partly depend on the area and depth within the brain or peripheral system which requires stimulation. Effected areas closer to the surface of the brain could be treated with Microarray LEDs [12] while deeper areas would have some delivery system consisting of fiber optic wires [22]. Finally any issues of plasticity, whether Hebbian or homeostatic [9] [10] must be addressed when repeatedly activating an ectopic cation channel and the consequent implication on rehabilitation.

4 Discussion and Conclusions

Problems associated with conventional electrical stimulation of neurons include:

- Electrical stimuli can excite action potentials but cannot inhibit neuron activity.
- It is difficult to confine electrical fields to target specific cell types or structures.
- Possible alterations in neural tissue due to power dissipation (specially for large arrays of stimulators).

Here we propose to use light activated channels as alternative approach for chronic stimulations, which could resolve some of these issues. In this context examples of optogenetic control of movement in rodents using fiber optic wires and ChR2 expression in the motor cortex have already been succesfully demonstrated [20]. Advances in the field of optogenetics have engineered new ChR2 versions making it possible to achieve sustained spike trains up to at least 200Hz [27]. In addition ChR2 has been mutated into a step-function opsin (SFO) making neurons in a state of increased exciteability after brief pulses of light which can then in turn be reversed by a pulse of yellow light [28].

However, it still remains to be seen if these tool and its advances can restore biologically relevant activity to effected brain areas as it has mostly been used as an exploratory tool. Recently the potential of this tool in rehabilitation was explored experimentally in muscle control. In movement rehabilitiation stimulation with electric cuffs so far has been disappointing. The reason for this is because it tends to recruit large, fatigable motor neurons and not the smaller motor neurons. The significance of this to paralyzed patients is that although they are able to walk when electrically stimulated, this is only sustainable for several minutes as larger fast twitch nerve fibers respond preferentially to the small-slow twitch nerves. As it is the smaller nerve fibers that contol refined movement their absence during activation results in jerky movement and rapid fatigue. Promisingly, optogenetic stimulation of mouse peripheral nerves seems to achieve the orderly recruitment of these muscles enhancing perfomance while concurrently reducing fatigue [29]. Should these experiments prove fruitfull the use of alternative light sensitive proteins which inhibit activity could help alleviate diseases in which hyperexciteability causes spasticity such as in cerebral pulsy or convulsion in epilepsy. Such a tool already exists in the form of the chloride pump, Halorhodopsin which is activated by yellow light [30]. The addition of this protein to the optogentic tool box now also permits for controlled inhibition of neural activity. This will be necessary when considering gabaergic inhibitory inputs and their role in neural circuits when trying to restore meaningful activity to neural networks.

Optogenetic tools have proved to be an invaluable exploratory method for probing neural network activity and plasticity [14] [19] and we are just entering an era were it may prove to be a very successful therapeutic tool.

Acknowledgements

This work has been supported in part by the ONCE (National Organization of the Spanish Blind), by the Research Chair on Retinitis Pigmentosa and by grant SAF2008-03694 from the Spanish Government.

References

1. Normann, R.A., Greger, B.A., House, P., Romero, S.F., Pelayo, F., Fernandez, E.: Toward the development of a cortically based visual neuroprosthesis. J. Neural Eng. 6(3). 35001 (2009)
2. Fernandez, E., Pelayo, F., Romero, S., Bongard, M., Marin, C., Alfaro, A., Merabet, L.: Development of a cortical visual neuroprosthesis for the blind: the relevance of neuroplasticity. J. Neural Eng. 2(4), R1–R12 (2005)
3. Fallon, J.B., Irvine, D.R., Shepherd, R.K.: Cochlear implants and brain plasticity. Hear. Res. 238(1–2), 110–117 (2008)
4. Awan, N.R., Lozano, A., Hamani, C.: Deep brain stimulation: current and future perspectives. Neurosurg. Focus 27(1), 2 (2009)
5. Marin, C., Fernandez, E.: Biocompatibility of intracortical microelectrodes: current status and future prospects. Front Neuroengineering 3, 8 (2010)
6. Cogan, S.F., Troyk, P.R., Ehrlich, J., Gasbarro, C.M., Plante, T.D.: The influence of electrolyte composition on the in vitro charge-injection limits of activated iridium oxide (AIROF) stimulation electrodes. J. Neural Eng. 4(2), 79–86 (2007)
7. Hascup, E.R., af Bjerken, S., Hascup, K.N., Pomerleau, F., Huettl, P., Stromberg, I., Gerhardt, G.A.: Histological studies of the effects of chronic implantation of ceramic-based microelectrode arrays and microdialysis probes in rat prefrontal cortex. Brain Res. 1291, 12–20 (2009)
8. McCreery, D.B., Yuen, T.G., Agnew, W.F., Bullara, L.A.: Stimulus parameters affecting tissue injury during microstimulation in the cochlear nucleus of the cat. Hear Res. 77(1-2), 105–115 (1994)
9. Matthew, G., Burrone, J.: Activity-dependant relocation of the axon initial segment fine-tunes neuronal excitability. Nature 465 (2010)
10. Goold, C.P., Nicoll, R.A.: Single-cell optogenetic excitation drives homeostatic synaptic depression. Neuron 68(3), 512–528 (2010)
11. Boyden, E.S., Zhang, F., Bamberg, E., Nagel, G., Deisseroth, K.: Millisecond-timescale, genetically targeted optical control of neural activity. Nat. Neurosci. 8(9), 1263–1268 (2005)
12. Grossman, N., Poher, V., Grubb, M.S., Kennedy, G.T., Nikolic, K., McGovern, B., Berlinguer Palmini, R., Gong, Z., Drakakis, E.M., Neil, M.A., Dawson, M.D., Burrone, J., Degenaar, P.: Multi-site optical excitation using ChR2 and micro-LED array. J. Neural Eng. 7(1), 16004 (2010)

13. Ranganathan, R., Harris, W.A., Zuker, C.S.: The Molecular Genetics of Invertebrate Phototransduction. Trends in Neuroscience 14, 486–493 (1991)
14. Zemelman, B.V., Lee, G.A., Ng, M., Miesenbock, G.: Selective Photostimulation of Genetically ChARGed Neurons. Neuron 33, 15–22 (2002)
15. Nagel, G., Ollig, D., Fuhrmann, M., Kateriya, S., Musti, A.M., Bamberg, E., Hegemann, P.: Channelrhodopsin-1: A Light-Gated Proton Channel in Green Algae. Science 296, 2395–2398 (2002)
16. Suzuki, T., Yamasaki, K., Fujita, S., Oda, K., Iseki, M., Yoshida, K., Watanabe, M., Daiyasu, H., Toh, H., Asamizu, E., Tabata, S., Miura, K., Fukuzawa, H., Nakamura, S., Takahashia, T.: Archaeal-type rhodopsins in Chlamydomonas: model structure and intracellular localization. Biochemical and Biophysical Research Communications 301, 711–717 (2003)
17. Nagel, G., Szellas, T., Huhn, W., Kateriya, S., Adeishvili, N., Berthold, P., Ollig, D., Hegemann, P., Bamberg, E.: Channelrhodopsin-2, a directly light-gatedcation-selective membrane channel. PNAS 100(24), 13940–13944 (2003)
18. Ishizuka, T., Kakuda, M., Araki, R., Yawo, H.: Kinetic evaluation of photosensitivity in genetically engineered neurons expressing green algae light-gated channels. Neurosci. Res. 54(2), 85–94 (2006)
19. Schroll, C., Riemensperger, T., Bucher, D., Ehmer, J., Voller, T., Erbguth, K., Gerber, B., Hendel, T., Nagel, G., Buchner, E., Fiala, A.: Light-Induced Activation of Distinct Modulatory Neurons Triggers Appetitive or Aversive Learning in Drosophila Larvae. Current Biology 16, 1741–1747 (2006)
20. Nagel, G., Brauner, M., Liewald, J.F., Adeishvili, N., Bamberg, E., Gottschalk, A.: Light Activation of Channelrhodopsin-2 in Excitable Cells of Caenorhabditis elegans Triggers Rapid Behavioral Responses. Current Biology 15, 2279–2284 (2005)
21. Bi, A., Cui, J., Ma, Y.-P., Olshevskaya, E., Pu, M., Dizhoor, A.M., Pan, Z.-H.: Ectopic Expression of a Microbial-Type Rhodopsin Restores Visual Responses in Mice with Photoreceptor Degeneration. Neuron 50, 23–33 (2006)
22. Aravanis, A.M., Wang, L.-P., Zhang, F., Meltzer, L.A., Mogri1, M.: An optical neural interface: in vivo control of rodent motor cortex with integrated fiberoptic and optogenetic technology. J. Neural Eng. 4, 143–156 (2007)
23. Hägglund, M., Borgius, L., Dougherty, K.J., Kiehn, O.: Activation of groups of excitatory neurons in the mammalian spinal cord or hindbrain evokes locomotion. Nature neuroscience 13(2) (2010)
24. Wang, H., Peca, J., Matsuzaki, M., Matsuzaki, K., Noguchi, J., Qiu, L., Wang, D., Zhang, F., Boyden, E., Deisseroth, K., Kasai, H., Hall, W.C., Feng, G., Augustine, G.J.: High-speed mapping of synaptic connectivity using photostimulation in Channelrhodopsin-2 transgenic mice. Proc. Natl. Acad. Sci. U. S. A. 104(19), 8143–8148 (2007)
25. Godley, B.F., Shamsi, F.A., Liang, F.Q., Jarrett, S.G., Davies, S., Boulton, M.: Blue light induces mitochondrial DNA damage and free radical production in epithelial cells. J. Biol. Chem. 280(22), 21061–21066 (2005)
26. Diester, I., Kaufman, M.T., Mogri, M., Pashaie, R., Goo, W., Yizhar, O., Ramakrishnan, C., Deisseroth, K., Shenoy, K.V.: An optogenetic toolbox designed for primates. Nature Neuroscience 14(3), 387–397 (2011)
27. Gunaydin, L.A., Yizhar, O., Berndt, A., Sohal, V.S., Deisseroth, K., Hegemann, P.: Ultrafast optogenetic control. Nat. Neurosci. 13(3), 387–392 (2010)
28. Berndt, A., Yizhar, O., Gunaydin, L.A., Hegemann, P., Deisseroth, K.: Bi-stable neural state switches. Nature Neuroscience 12(2) (February 2009)

29. Llewellyn, M.E., Thompson, K.R., Deisseroth, K., Delp, S.L.: Orderly recruitment of motor units under optical control in vivo. Nature Medicine 16(10) (October 2010)
30. Zhang, F., Wang, L.-P., Brauner, M., Liewald, J.F., Kay, K., Watzke, N., Wood, P.G., Bamberg, E.: Multimodal fast optical interrogation of neural circuitry. Nature 446 (April 2007)

Classification Tree Generation Constrained with Variable Weights*

Pedro Barahona[1], Gemma Bel-Enguix[2], Veronica Dahl[2,3],
M. Dolores Jiménez-López[2], and Ludwig Krippahl[1]

[1] Departamento de Informática, Universidade Nova de Lisboa
{pb,ludi}@di.fct.unl.pt
[2] Research Group on Mathematical Linguistics, Universitat Rovira i Virgili
{gemma.bel,mariadolores.jimenez}@urv.cat
[3] Department of Computer Science, Simon Fraser University
veronica@cs.sfu.ca

Abstract. Trees are a useful framework for classifying entities whose attributes are, at least partially, related through a common ancestry, such as species of organisms, family members or languages. In some common applications, such as phylogenetic trees based on DNA sequences, relatedness can be inferred from the statistical analysis of unweighted attributes. In this paper we present a Constraint Programming approach that can enforce consistency between bounds on the relative weight of each trait and tree topologies, so that the user can best determine which sets of traits to use and how the entities are likely to be related.

1 Introduction

Classification trees are useful methods of research in many areas, including biology and linguistics, even though the characteristics of the problems may differ. In biology, for example, distances between genes are clearly identifiable, and they very strongly correlate with phylogeny, since most mutations are evolutionary neutral [5]. This allows phylogenetic trees to be built using hierarchical clustering methods with fixed distances [3]. However in linguistics, distance measures based on arbitrary sets of features may not correlate with phylogeny because many differences are not random.

Even in biology, in some cases it is not possible to measure the genetic information, such as when studying fossils, and one must rely on anatomical features which may not correlate with phylogeny as well as the mostly random inherited mutations in ADN. For these cases, the assumption that all observable features have the same importance in determining phylogeny is likely not to hold. Distance between features can therefore vary according to the relevance that is assigned to each feature, leading to different classification trees being generated by hierarchical clustering.

* This paper has been supported by the project HP2008-0029.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 274–283, 2011.

The number of such classification trees grows very fast with the number of objects in the classification, making it crucial to have an efficient method for eliminating uninteresting options.

In this article we propose a novel methodology for constraining the number of classification trees through allowing different features to be weighted with some slack with respect to an average. Our methodology, which can be flexibly applied to multiple uses both for linguistics and for biology, drastically diminishes the number of alternatives to be considered, down to an acceptable number, by adopting a constraint programming approach. It may handle arbitrary sets of features, and uses an implicit distance measure while allowing for an arbitrary flexibility in the relative weights of the features. The purpose of this approach is to work dually, by finding either: a) the trees that are possible given some bounds on the weights, or b) the bounds on the weights given constraints on possible trees. This is achieved either by specifying allowed trees or simply the constraints that stipulate the tree topology and discovering whether there are sets of weights that make the distances compatible with such topologies.

Section 2 briefly describes the state of the art. Section 3 describes the motivation for this paper, the main features of Constraint Programming and, in subsection 3, our phylogenetic tree model; section 4 shows the results obtained for a biological and a linguistic example, and section 5 discusses our results.

2 Background, State-of-the-Art

Phylogenetic trees aim to represent the evolutionary relationships between entities, either qualitatively, by grouping together those that are more closely related, or quantitatively, with each branch length indicating the distance to a common ancestor, and the path through the branches of the tree the distance between any two entities.

There are essentially two different ways of calculating phylogenetic trees. Distance methods rely on distance values between all pairs of entities to organize in the tree [3]. The tree is then calculated such that the sum of the branches joining any pair equals their corresponding distance. Commonly used algorithms to build phylogenetic trees based on distance measures between the nodes (almost always the leaves of the tree) are the Unweighted Pair Group Method with Arithmetic Mean (UPGMA) and the neighbor-joining method [8].

Distance methods tend to be the first choice, as sequence comparisons allow for distances to be pre-computed. However, for some applications, it is necessary to use sets of features from which a distance measure cannot be immediately inferred.

3 Our Proposed Methodology

We use a declarative, CLP, algorithm for building the trees by specifying the constraints that stem from the evolutionary relationships that the tree must represent. Each inner node bisects the set of entities that descend from the ancestor represented in that node. From that node there are two branches, each one the origin of a sub-tree leading to a set of leaf entities. Let A and B be those sets. Evolutionary relations require that all pairs of entities in A have a closer kinship than any pair formed by one entity from A and one entity from B. This constraint in the evolutionary relations of these groups is what the branching represents. More formally:

$$\forall a_i, a_j \in A, \forall b_k \in B, dist(a_i, a_j) < dist(a_i, b_k) \tag{1}$$

This results in a top-down computation of the phylogenetic tree, requiring, at each inner node, the calculation of a bisection of the set of entities in that subtree. Since there are 2^n bisections of any set of n elements, this is potentially an exponential problem unless there are some ways of reducing the number of possibilities. This approach has some similarity to top down induction of decision trees [7] or clustering trees [1], but our focus is not so much on the generation of the tree itself, or the efficiency of that step, but rather on the flexibility of the distance measures. Thus, for this purpose, we can consider the actual implementation details of the tree computation may be adapted according to the size and complexity of the problems to solve.

The distance between two entities is a linear combination of the modulus of the difference of the feature values. However, the weights are not instantiated, but have an initially specified domain that can be constrained by the tree topology. Considering S to be the set of all entities to organize in the phylogenetic tree, the distance between any to entities is defined as

$$\forall s_i, s_j \in S, dist(s_i, s_j) = \sum_{k=1}^{a} w_k \times |f(s_i, k) - f(s_j, k)| \tag{2}$$

where $f(s_i, k)$ is the attribute of index k for the entity of index i (respectively for $f(s_j, k)$) and w_k is the weight of attribute k. The attribute weights are also constrained to be greater or equal to zero and to add up to a constant value, which we can assume here to be the unit value (although the actual value is arbitrary). Thus:

$$\forall w_k \in W, w_k \geq 0 \tag{3}$$

$$\sum_{k=1}^{a} w_k = 1 \tag{4}$$

Moreover, if all features are normalised so that the maximum difference between any two values of the same feature is 1, then the maximum distance between two leaves is 1 (assuming the sum of the weights to be 1, the value that we have set).

For our proof-of-concept exercises, we consider the additional constraint that the weight of any attribute must fall within a preset interval, that the user can define. If the lower bound is zero, then that attribute can be ignored. The particular case where the weights are restricted to the $1/N$, with N being the number of weights used, results in the computation of an unweighted tree, if one such tree is possible.

These parameters constrain all trees to respect the distance constraints imposed by the tree metric, with weights that fall within the preset bounds, which requires posting $O(n^3)$ "less than" constraints, which are easily maintained by a linear constraint solver. Under the constraint programming paradigm, most trees are eliminated in the earlier stages of their top down recursive generation if these constraints are impossible to satisfy.

Furthermore, given the declarative nature of our approach, the tree typology may be completely or partially specified, in which case the system is used to test whether there are attribute weights consistent with given phylogenetic relations. Finally, for each tree that satisfies the constraints, a specific set of weight values is calculated that minimizes the range of all weights.

4 Examples from Biology and Linguistics

To illustrate our approach, we give two application examples, as proof-of-concept exercises. One is a study of the evolutionary relationships in Eocene primate fossils, using two subsets of eight and nine fossil specimens from [9], characterized by a selection of 26 features that had numerical values and were all defined for these 15 fossil species. The other example is the phylogenetic classification of languages, using a set of six Latin languages, two Germanic languages and Basque, characterizing each language with a set of twenty phonetic and grammatical features.

4.1 Biological Examples

For these case studies we chose two different applications. In one case, we assume that the correct phylogenetic tree is known from independent sources, but we wish to assess the suitability of a set of features for grouping our entities according to that tree. A real-life scenario where this may occur is in the classification of fossil specimens. For a set of fossils, the evolutionary relations may be clear from geological and chronological considerations or because the fossil specimens are particularly well preserved and rich in detail. However, in another set of specimens the reverse may happen, and it may be necessary to assess if the features that can be observed in this second group are sufficient for computing the phylogenetic tree and, if so, what should their relative weights be.

To illustrate this scenario, we considered a small subset of the primate fossil data available from [9]. Fig. 1 shows the phylogenetic relations between species in two parts of the published phylogenetic tree, which we assume to correctly represent their phylogenetic relations. In the original paper, the authors used a set of 360 fossil traits to classify a total of 117 species. For our prototype implementation, this data set would be too complex, not only due to its size but also because, for each species, it is often the case that a significant fraction of the 360 features are missing, since fossil quality varies significantly between diferent specimens. So we selected these smaller subsets, A and B, of eight and nine species, respectively, and used only 26 features for which the data was complete in all 17 species, and which were identical in all of these species.

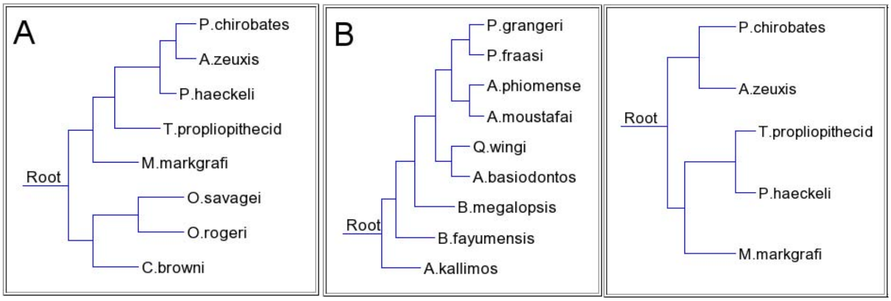


Fig. 1. The two groups of fossil primate species, A and B, and their respective phylogenetic trees used in our bioinformatics examples. The rightmost panel shows the uniform weights tree that can be calculated from a subset of 5 elements from A. Original data from [9]. Tree representations drawn with T-Rex [6].

The third panel of Fig. 1 shows the tree obtained assuming that all features have the same weight. This uniform-weight tree is only possible for a subset of 5 species from the A set, as the full set of 8 species from A cannot be organized into a uniform-weight tree, with these 26 features that we used. However, even with only 5 elements, the tree produced with uniform weights is incorrect, grouping *T. propliopithecid* with *P. haeckeli.* To reproduce the correct trees for A, B and event the 5 species subset of A, it is necessary to allow feature weights to vary considerably. Whereas with a uniform weight distribution each feature contributes 3.85% of the total, the correct tree for A requires weights to vary between 0.84% and 5.11%. The tree for the B group is even more demanding, with four features needing to be above 14% of relative weight each, while eleven have a weight of 0%, meaning they must be completely discarded for the tree to be possible. There are not unique solutions to the problem of building a tree, but they are the solutions found by minimizing the range of relative weights assigned to features, and thus are the solutions closest to the uniform-weight metric, in this sense.

If species are sufficiently related, it is possible to extrapolate from the weights of one to the other. For example, the weights used to form the correct tree for the subset of 5 species of A have a correlation of 0.73 to the weights in the correct tree of the full A set, and many are quite similar, as shown in Fig. 2.

If the groups are too different, this extrapolation is not possible. When comparing the B and A groups, which were chosen from different regions of the larger tree containing all 117 fossil species reported in [9], the correlation of feature weights is merely 0.3, and the scores differ markedly between the two trees. However, if only two species from B are added to A, for a ten-species tree, this tree can be correctly computed with only small differences in feature weights, and a correlation of 0.72. Fig. 3 shows this comparison.

Though these are still somewhat preliminary results, they show two important aspects of this problem. First, it is often impossible to compute the correct phylogenetic trees if one has incomplete data and assumes uniform weights across all features. In fossil analysis, rich datasets like those in [9] are the exception rather

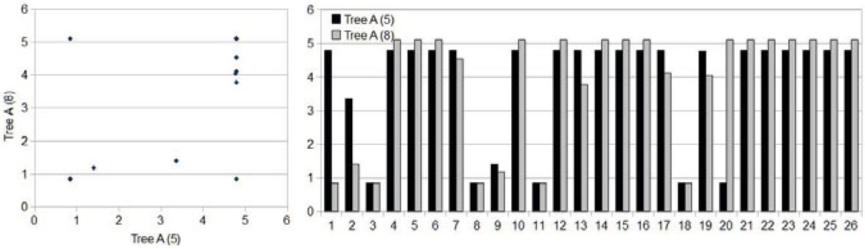


Fig. 2. Comparison of the relative weights in the correct trees for the complete *A* group and for the first five species of *A*, *Tree A (5)* and *Tree A (8)* respectively. The left panel shows a cross-plot of the weights. Values are in percentage.

the norm, but are necessary if one needs to overcome by statistical averaging the problem of different features having different impacts in phylogeny. In general, it is hard to collect data on hundreds of different features in fossilized organisms. Furthermore, it is possible, at least in some cases, to assign weights to features on the basis of a known phylogenetic tree and then extend those metrics to include additional organisms for which data may be highly incomplete. This raises the possibility of extending the scope of these computations to cases where, until now, data was insufficient to support a phylogenetic tree.

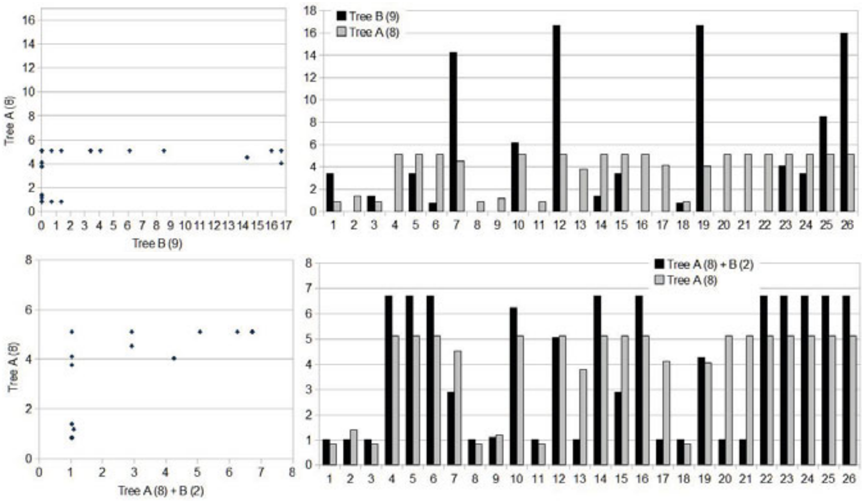


Fig. 3. Comparison of the relative weights in the correct trees for the *B* and *A* groups (*Tree B (9)* and *Tree A (8)* respectively, top panels) and for the *A* group plus the two first species of *B*, compared to the weights of the *A* group (*Tree A (8) + B (2)* and *Tree A (8)*, bottom panels). The left panels shows the cross-plots of the weights. Values are in percentage.

4.2 Linguistic Examples

In the last years, many authors have stressed the close relationship between the genetic tree of the human species and the phylogenetic tree of human languages [2]. It seems that language evolution and the evolution of species are very related, and this allows us to try to approach both processes with the same tools. However, there is an important difference between them. Whereas the genetic code of two different species does not cross once separate, the influence of a language on others because of geographical, political or social reasons is constant in history. In spite of this, we think that our constraint-based method can help the researchers to establish a hierarchical ordering of features in languages, starting from the data we know. In the future, an improved version of the method, dealing with an enough representative number of features could help to understand the processes of derivation and contact of languages that are not well-known so far.

Classification of natural languages has been performed by considering as basis different underlying principles. In general, it is possible to distinguish the following three directions: a) *Genetic Classification* that pays attention to the historical evolution of languages; b) *Typological Classification.* In this type of linguistic research the classification is obtain by analysing the internal structure of languages; c) *Areal Classification.* This type of classification considers geographical closeness and contacts between languages. The different classifications do not match each other. However, the correlation between them may be of importance for linguistic research. We claim that our methodology can contribute to the three different types of classifications. In what follows, we illustrate this idea with several examples. For our linguistic 'toy example' we have chosen a limited set of 9 languages: Basque (BAS), German (GER), English (ENG), Portuguese (POR), Spanish (SPA), Catalan (CAT), French (FRE), Italian (ITA) and Romanian (ROM). We have selected, in an almost-aleatory way, twenty different features, gathered in four groups:

1. Phonetics: a) nasal consonants, b) voiced and voiceless fricative dentals; c) voiced and voiceless fricative post-alveolars; d)double consonants; e) distinction voiced and voiceless fricative labiodentals; f) number of vowels.
2. Morphology: a) number of morphological genders; b) phrase agreement; c) Verbal inflection; d) case; e) plural formation.
3. Syntax: a) partitive Pronoun; b) precedence Noun Adjective; c) SVO order; d) ergativity; e) obligatory Subject; f) *to be* as auxiliar for the past tenses; g) auxiliary word for negation.
4. Lexicon: a) days of the week; b) numbers from 10 to 20.

This distribution is well balanced between Phonetics, Morphology and Syntax, and gives a small prominence to Lexicon, the most fluctuant part of language. Some of these features have been assigned a 0/1 distribution, and some others, a distribution with a greater range, later normalized.

Intuitively, it is easy to see that, giving to every feature the same weight, it is going to be a hard task to complete a consistent tree. However, the weights can help the system to build a 'correct' tree. As an example, ergativity is a very non-common feature of languages. Only a very small set of non-indoeuropean languages are ergative. Thus, assigning a high weight to this feature is very convenient to be able to distinguish Basque from the other languages and, consequently, correctly generate the first leaf of the tree.

Regarding latin languages, they have been classified in western and eastern languages with a criterion based on phonetic features, more than 'phylogenesy'. The probably best way to group these languages by filiation would be to relate all of them to Latin in the same way. Moreover, we know that the Romania is a linguistic continuum, and it is difficult to find the key features to define a consistent gathering. In spite of this, many authors would consider a classification like the one in Fig. 4 as the most closer to the correct one.

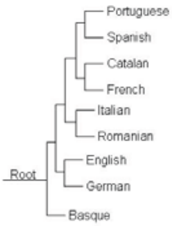


Fig. 4. 'Classical' derivation tree of the languages in the example

Such taxonomy has an additional problem: the grouping mixes the genetic, typogical and areal criteria explained above. While the distinction between Germanic and Latin languages is genetic, the distinction between the different romance languages is typological. Some different classifications could be obtained depending on the methodology and the features taken. On the other hand, the results that Basque obtains, very close to Spanish, are mainly because of areal reasons, due to the fact that history has made these two very different languages to coevolve in many features.

The system is challenged to obtain a tree similar to Fig. 4 without the inference of the intermediate languages. Firstly, we focus in trying to obtain the tree of Romance Languages with no restriction. The program generates two different trees.

The best tree is shown in panel A of Fig. 5. This tree leaves Romanian aside and groups Spanish with Portuguese and Catalan with Italian. The weights it requires are in a range of 3.66-5.88. To build this tree, the system needs to decrease the value of the following features: 1(c), 1(d), 1(e), 1(f), 3(e), 3(f), 3(g). These features are phonetic and syntactical, and they are around 3.66.

The second proposed tree is shown in panel B of Fig. 5. This tree moves Romanian closer to Spanish and Portuguese, and leaves Catalan and Italian in the other group, with the French aside. However, this option requires a larger range in the weight values: 3.5-6.98. Here, the traits that had their weight increased were: 1(c), 1(d), 1(f), 3(a), 3(d), 3(f), 3(g), 4(a).

Nevertheless, the variation in the weights only produces the change in the position of Romanian, what means that, in general, the balanced distribution of the selected features produces the correct trees for Roman languages. All this can be seen in Fig. 5.

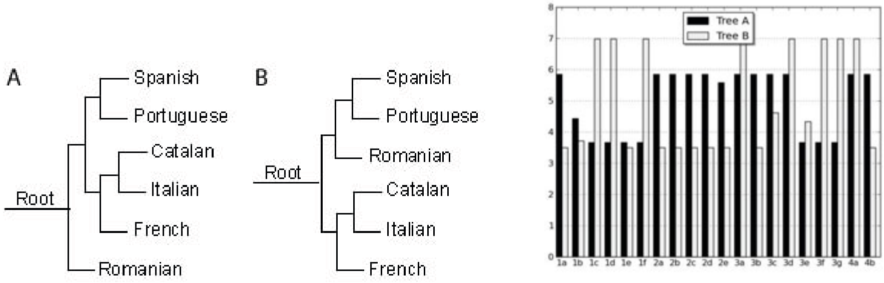


Fig. 5. A and B are the trees the system is able to obtain with the 6 latin languages considered in the example. Right: comparison in the variation in the weights the system needs to perform to obtain the trees.

However, the system can be forced to find the relevant features to produce a tree with some constraints. For example, following the classical distribution of Latin languages we can ask the program to find a tree enforcing Italian and Romanian together. It is possible to obtain a tree like the one shown in Fig. 6, with the weights in the range of 2.14-7.98. The traits with the largest weights are 1(f), 2(c), 2(e), 3(b), 3(c), 3(d), 3(e), 3(f), 3(g), among which are features that closely relate Romanian and Italian, like the number of vowels of their respective systems (7) and the plural formation. But it is unclear why the system needs to increase the value of the feature ergativity, that is shared by every Romance language, except perhaps as a neutral way of reducing the relative weight of all other features. The graphical representation of the result is found in Fig. 6.

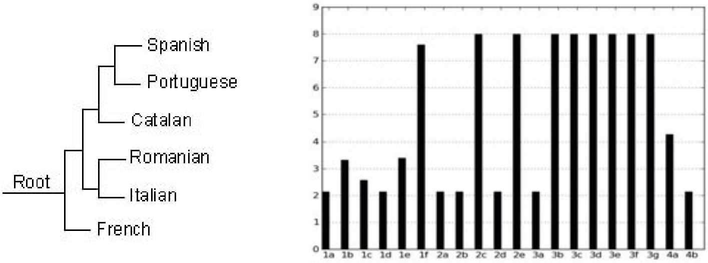


Fig. 6. Tree obtained by forcing the system to produce together Romanian and Italian. Plot of the variation of the weights.

5 Conclusion

In this paper we presented a hierarchical classification methodology that is able to generate or test classification trees of entities whose features may have different importance. This is achieved by adopting a distance function between the entities where the contribution of the different features can be weighted. Despite the potentially combinatorial number of trees, our constraint based method allows discarding most of them and generate an acceptable number when the weights are adequately constrained. We have also discussed the computational complexity of our method and shown that it can be applied to a moderate large number of entities. These characteristics were illustrated with examples from biology and linguistics, but the method is quite general and can be used in any domain, namely in situations where the classification of a set of entities depends on the relevance of a number of features that represent them, allowing the users to establish such relevance.

References

1. Blockeel, H., De Raedt, L., Ramon, J.: Top-Down Induction of Clustering Trees. ICML, 55–63 (1998)
2. Cavalli-Sforza, L.L., Feldman, M.: Cultural Transmission and Evolution. Princeton University Press, Princenton (1981)
3. Farris, J.S.: Estimating phylogenetic trees from distance matrices. Am. Nat. 106, 645–668 (1972)
4. Fitch, W., Margoliash, E.: Construction of Phylogenetic Trees. Science 155(3760), 279–284
5. Kimura, M.: Evolutionary rate at the molecular level. Nature 217(5129), 624–626
6. Makarenkov, V.: T-Rex: reconstructing and visualizing phylogenetic trees and reticulation networks. Bioinformatics 17, 664–668 (2001)
7. Quinlanm, J.R.: Induction of Decision Trees. Mach. Learn. 1(1), 81–106 (1986)
8. Saitou, N., Nei, M.: The Neighbor-joining Method: A New Method for Reconstructing Phylogenetic Trees. Mol. Biol. Evol. 4(4), 406–425 (1987)
9. Seiffert, E., et al.: Convergent evolution of anthropoid-like adaptations in Eocene adapiform primates. Science 461, 1118–1121 (2009)

Arithmetical Metabolic P Systems

Rosario Lombardo and Vincenzo Manca

Department of Computer Science, University of Verona
Strada Le Grazie, 15 – 37134 Verona, Italy
{rosario.lombardo,vincenzo.manca}@univr.it

Abstract. Metabolic P systems, a class of P systems introduced for modeling metabolic processes, proved to be promising in many contexts from bio-modelling to function approximation. In this paper we define a deterministic computational model based on this systems.

1 Introduction

Metabolic P systems represent metabolic processes in a discrete mathematical setting. Based on Gh. Păun's membrane computing theoretical framework [13], MP systems [6,7,8] are deterministic mono-membrane [2] dynamical systems where multiset rewriting rules are regulated by functions. In fact, biochemical reactions occur at a specific rate, determined by the *regulator* functions rather then selected with non-deterministic strategies. The encouraging results of applying MP systems to modeling biological phenomena and to function approximation [10] inspired the idea of defining a deterministic model of computation, called Arithmetical MP Systems, accompanied by a suitable software framework.

Since the P systems were introduced [12], several variants of the model were defined and many software application have been used. The P-Lingua [5] programming language, presented to be standard for non-deterministic P system models, supports the definition of a certain number of P system models offering a procedural calling of modules. The P-Lingua implementation simulator then runs one of the possible computations that the P system model may follow [3].

Arithmetical MP Systems introduce a deterministic model of computation for biological networks which can be composed together by defining a *sequential composition* among these systems. In Arithmetical MP systems the idea of sub-routine calling is not a procedural call but rather a *regulatory composition* mechanism allowing to regulate reaction rates by using the computational output of an adjacent system. Lamp, the preliminary framework implementing the language and a vast repertoire of mechanisms for assembling MP Systems, is available online at http://profs.sci.univr.it/~lombardo/lamp.

In MP systems, a multiset rule like $2a + b \rightarrow c$ means that a number u of molecules of kind a and the same number u of molecules b are replaced by u molecules of type c. The value of u is the *flux* of the rule application. Assume to consider a system at some time steps $0, 1, 2, \ldots, t$ and consider a substance x

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 284–294, 2011.

that is produced by rules r_1, r_3 and is consumed by rule r_2. If $u_1[i]$, $u_2[i]$, $u_3[i]$ are the fluxes of the rules r_1 , r_2 , r_3 respectively, in the transition from step i to step $i+1$, then the variation of substance x is given by

$$x[i+1] - x[i] = u_1[i] - u_2[i] + u_3[i] \ .$$

In a MP system, for any state, the flux of each rule is provided by a function, called *regulator* of the reaction. Substances, reactions, and regulators (plus parameters which are physical variables different from substances occurring as arguments of regulators) specify a dynamics, given at discrete steps. At every step, the dynamics is governed by a partitioning of mass determined by the fluxes of rules consuming it. An MP system is described entirely by an *MP grammar* where multiset rewriting reactions are given with the corresponding regulators. An MP grammar is also specified by an MP graph where the relationships between reactions and regulators appear in a direct way (see Fig. 1).

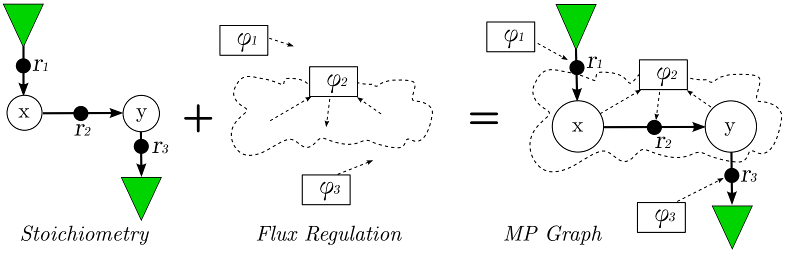


Fig. 1. MP graphs define the stoichiometric and regulatory components: *substances* (circles), *input* and *output* gates (green triangles) are connected by reactions arrows. The *regulators* (rectangles) determining the flux of rules are linked with dashed arrows.

2 Arithmetical MP Systems

The Arithmetical MP Systems are a class of MP Systems describing a universal computational model. Universality can be easily proved modelling a register machine or, with a little more effort, the μ–recursive functions.

Notational Conventions. A set of substances S, considered in a conventional order, determines a vector $X = (x_1, \ldots, x_n) \in \mathbb{N}^n$ of substance quantities providing the metabolic state of the system. When there is no possibility of ambiguity, for any j, $1 \leq j \leq n$, we use the notation x_j as substance quantity rather then substance symbol.

Let R be the set of reactions having the set of regulators $\Phi = \{\varphi_r \mid r \in R\}$. A reaction r is then represented by a left vector $r^- \in \mathbb{N}^n$ of reactants and a right vector $r^+ \in \mathbb{N}^n$ of products. The component-wise difference $r^\# = r^+ - r^-$ is the *stoichiometry balance* of r.

Definition 1. (x-availability constraint) *Let $x \in S$, if $R^-(x)$ is the subset of reactions consuming x, then x verifies the x-availability constraint with respect to Φ when*

$$\sum_{r \in R^-(x)} |r^-|_x \, \varphi_r(X) \leq x \ .$$

Where $|r^-|_x$ is the multiplicity of x in r^- using the Parikh notation.

Definition 2. (Blocked rule) *Let R be a set of reactions with regulators Φ. If, for a given substance x, the x-availability constraint is not satisfied with respect to Φ, then the reactions of $R^-(x)$ are said to be blocked on x with respect to Φ.*

Definition 3. (Simple regulators) *A set of regulators $\Phi = \{\varphi_r \mid r \in R\}$ is simple when, given the state $X = (x_1, \ldots, x_n)$,*

$$\forall r \in R, \quad \varphi_r(X) = \begin{cases} k \in \mathbb{N} & \textit{(constant)} \\ \textit{or} & \\ x_j \mid 1 \leq j \leq n & \textit{(projection)} \ . \end{cases}$$

Definition 4. (Arithmetical MP System) *An Arithmetical MP System of type (n, m, k) having n substances, m reactions and k parameters, is specified by the construct:*

$$M = (S, R, H, \Phi, \textsc{In}, \textsc{Out}, \textsc{Start}, \textsc{Halt})$$

where:

S *is a set of n substances, considered in a conventional order, determining the metabolic state $X \in \mathbb{N}^n$ of the system.*
R *is a set of m reactions $(r_1^-, r_1^+), \ldots, (r_m^-, r_m^+)$, whose stoichiometric balances are the columns of the stoichiometric matrix $\mathbb{A} = (r_1^{\#}, \ldots, r_m^{\#})$.*
H *is a function $H : \mathbb{N} \to \mathbb{N}^k$ providing, at each step $i \in \mathbb{N}$, the vector $H[i]$ of parameters.*
Φ *is the set of simple regulators $\{\varphi_r \mid r \in R\}$ such that, $\forall X \in \mathbb{N}^n$ each reaction $r \in R$ has associated the regulator:*

$$\widehat{\varphi}_r(X) = \begin{cases} 0 & \textit{if } r \textit{ is blocked w.r.t. } \Phi \\ \varphi_r(X) & \textit{otherwise} \ . \end{cases}$$

In, Out *are a set of substances with* In $\subseteq S$, Out $\subseteq S$.
Start, Halt *are two substances with* Start $\in S$, Halt $\in S$.

Let $X[0] \in \mathbb{N}^n$ be the initial state of the system M, its dynamics is specified by the vector recurrent equation, called Equational Matabolic Algorithm, defined in MP Systems [7]:

$$X[i+1] = \mathbb{A} \times \widehat{U}[i] + X[i] \qquad (1)$$

providing in $X[i+1]$ the metabolic state of the system by means of the vector of fluxes $\widehat{U}[i] = (\widehat{u}_r[i] \mid r \in R)$ where $\widehat{u}_r[i] = \widehat{\varphi}_r(x_1[i], \ldots, x_n[i], p_1[i], \ldots, p_k[i])$ are computed upon the components of the status $X[i]$, $H[i]$.

Arithmetical MP Systems compute functions according to the following:

Definition 5. (MP–Computable function) *A function* $f : \mathbb{N}^k \rightarrow \mathbb{N}^h$ *such that* $f(a_1, \ldots, a_k) = (b_1, \ldots, b_h)$ *is MP–computable if there exist an Arithmetical MP System* M *in which, when the* IN *substances contain the values* $a_1, \ldots, a_k$ *respectively, setting the* Start *substance to 1,* M *eventually ends having 1 in its* Halt *substance and the values* $b_1, \ldots, b_h$ *in the* OUT *substances.*

Moreover, M *never reaches the value 1 in the* Halt *substance if and only if* $f(a_1, \ldots, a_k)$ *is undefined.*

The Equational Metabolic Algorithm (1), expressed in algebraic terms in Def. 4, can also be expressed as multiset operations. Given the substances a, b, $c \in S$, we use the *polynomial representation* for denoting the multisets α, β, γ such as $\alpha = [\![n_a a + n_b b + n_c c + \ldots]\!]$, where $n_a, n_b, n_c, \ldots$ are the multiplicities for each molecular species. The multiset sum is easily defined summing component by component as, for example, $[\![a + 2b]\!] + [\![3b + c]\!] = [\![a + 5b + c]\!]$. The limited subtraction is analogously defined only if the subtrahend is less or equal then the subtracter, while the multiplication of a number times a multiset follows the distributive property of polynomials as for example $3 \times [\![a + 2b]\!] = [\![3a + 6b]\!]$.

Given a reaction $r : 2a + b \rightarrow c$, we can naturally read its reactants and products as multisets, using again with some abuse of notation, $r^- = [\![2a + b]\!]$ and $r^+ = [\![c]\!]$ respectively. Now we can use the multiset operations to express the Equational Metabolic Algorithm (1), whereas, at each step $i \in \mathbb{N}$, the rule r subtracts $\widehat{\varphi}_r(X[i])$ times the multiset r^- and adds $\widehat{\varphi}_r(X[i])$ times the multiset r^+ to the metabolic state $X[i]$:

$$X[i+1] = X[i] + \left(\sum_{r \in R} \widehat{\varphi}_r(X[i]) \times r^+ - \sum_{r \in R} \widehat{\varphi}_r(X[i]) \times r^- \right) \quad . \tag{2}$$

2.1 Simple Arithmetical MP Systems

In this section some introductory *MP–computable functions* are described in order to get the reader acquainted with the Arithmetical MP Systems, MP grammars and MP graphs. In MP graphs the flux regulators are written besides the rule nodes and the graphical notations of Table 2.1 will be used.

Limited Subtraction. The limited subtraction, written as $x \ominus y$, is properly defined only if the subtrahend can actually be subtracted from the subtracter. More formally:

$$\ominus : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}, \qquad x \ominus y = \begin{cases} x - y \text{ if } x \geq y \\ x \text{ otherwise } . \end{cases}$$

The Arithmentical MP system implements the operation in one step as shown in Fig. 2, where the rule tries to expel x and y with a flux equals to the subtrahend

Table 1. Graphical notations

Symbol	Description
○↖	IN substances
○↘	OUT substances
○⤡	IN–OUT substances
◎	START substance
⊙	HALT substance
▽↓	open system input rule $\lambda \to a$
↓▽	open system output rule $a \to \lambda$
○	substance
▭	membrane

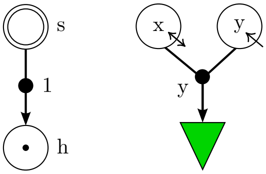


Fig. 2. Limited Subtraction. Note that x contains part of the input and the output.

y. If there is not enough of both molecular species the rule blocks and the input will be left untouched. Otherwise a proper subtraction $x - y$ will be left in x.

Note that the rules are unlinked from the START/HALT signalling substances, that is, the system needs the IN substances to be fed at the same time with the START substance. Moreover, after the system has provided the result, the IN/OUT substances (x, y) are likely to be left in state which does not allow to safely start another computation.

The definition of the MP-Limited-Subtraction is:

$$\begin{aligned} S &= \{x, y, s, h\} \\ \textsc{In} &= \{y\} \\ \textsc{In/Out} &= \{x\} \\ \textsc{Start} &= s \\ \textsc{Halt} &= h \end{aligned} \qquad \overbrace{\begin{aligned} &r_1 : s \to h \\ &r_2 : xy \to \emptyset \end{aligned}}^{R} \quad \overbrace{\begin{aligned} \varphi_1 &= 1 \\ \varphi_2 &= y \end{aligned}}^{\Phi}$$

Conditional Construct. The Conditional construct is implemented addressing the notes mentioned above: the whole network is linked to the START/HALT signalling substances and there are clean-up reactions making the system slightly more complicated but also *reusable*. Given two alternative biological MP networks, one must be chosen depending on the value of a guard. The idea is therefore to produce two alternative substances used to START the corresponding system (see Fig. 3). Encoding the truth values {*true*, *false*} as {1, 0} we are defining the function:

$$\mathrm{Cond} : \mathbb{N} \mapsto \{0,1\} \times \{0,1\}, \qquad \mathrm{Cond}(z) = \begin{cases} (1,0) & \text{if } z = 0 \\ (0,1) & \text{otherwise} \end{cases} .$$

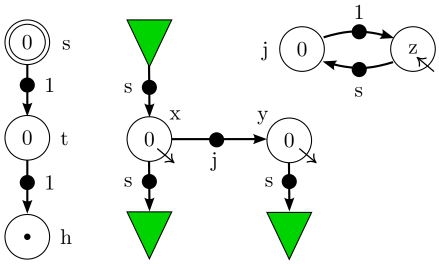


Fig. 3. The conditional construct is a function $\mathbb{N} \mapsto \{0,1\} \times \{0,1\}$

The following MP grammar defines the Conditional construct:

$$\begin{array}{rl} & \\ S = \{x, y, z, j, t, s, h\} & \\ \textsc{In} = \{z\} & \\ \textsc{Out} = \{x, y\} & \\ \textsc{Start} = s & \\ \textsc{Halt} = h & \end{array} \quad \overbrace{\begin{array}{l} r_1 : z \to j \\ r_2 : j \to z \\ r_3 : \emptyset \to x \\ r_4 : x \to y \\ r_5 : x \to \emptyset \\ r_6 : y \to \emptyset \\ r_7 : s \to t \\ r_8 : t \to h \end{array}}^{R} \quad \overbrace{\begin{array}{l} \varphi_1 = s \\ \varphi_2 = 1 \\ \varphi_3 = s \\ \varphi_4 = j \\ \varphi_5 = s \\ \varphi_6 = s \\ \varphi_7 = 1 \\ \varphi_8 = 1 \end{array}}^{\Phi}$$

Once the Start s becomes 1, x is incremented to 1 from the environment and, at the same time, $r_1 : z \to j$ tries to increase the value of j from 0 to 1 with a flux of $s = 1$.

Input $z = 0$: the rule $r_1 : z \to j$ blocks and the value of j remains 0. In the next step the rule $t \to h$ signals the end of computation and the result in Out is $(x, y) = (1, 0)$.

Input $z \neq 0$: the value of j actually changes to 1 and in the next step the rule $r_4 : x \to y$ inverts the values of x and y, leading to a final configuration where the Out is $(x, y) = (0, 1)$.

The reactions $r_5 : x \to \emptyset$ and $r_6 : y \to \emptyset$ clean up the Out substances at the beginning of any new computation. The rule $r_2 : j \to z$, restores the In substance to its original value, if needed. However the ancillary substance j is not introduced as much for the restoration of x but rather for its two properties. First, it is the complementary of input z when needed to invert the output and, second, being always zero at the beginning, the rule $r_4 : x \to y$ never blocks the cleaning of $r_5 : x \to \emptyset$. Finally the temporary t is introduced for timing purposes suggesting that the system always takes two steps to execute.

3 Adding Membranes

In the examples above there was the implicit assumption that the IN values were available at once and together with the Start. Indeed biological networks transform matter as soon as appropriate rules and substances are put together. If we want to compose two computation processes together, a sort of buffering layer is needed in-between preventing any partial values being fed into the second system possibly corrupting its initial configuration and triggering altered computations. We want to be sure that such a composition is possible at all.

Theorem 1. (Sequential Composition) *Given two MP Computable functions f and g, the composition $f \circ g$ is a MP Computable function.*

A mechanism proving the theorem is shown in Fig. 4 where top and bottom substances belong to f and g respectively. Let's consider, without loss of generality, only the output x' of f being transformed into the input x for g.

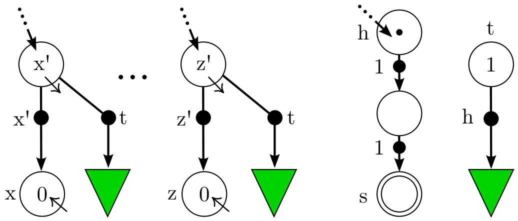


Fig. 4. The Sequential Composition mechanism

Anytime before h reaches 1, both the rules consuming x' are *blocked* because the total requested flux $x'+t$ exceeds the value of x'. Once h reaches 1, substance t gets its value decreased to 0 and the x'–availability constraint is satisfied: Start and In substances are synchronously provided to g. The Start and Halt substances of $f \circ g$ are the Start of f and the Halt of g, respectively.

3.1 Multicompartimental Setting

The functional composition, a very desirable property when complex algorithms are to be designed, naturally brings the idea of having separate and composable computing entities: the membranes.

A boundary rule [1] is *localized* in the membrane of its reactants. Let k, l be regions, a boundary rule has always the form of *localized*$_k$ *reactants* $\rightarrow$ *localized*$_l$ *products*. If the regions k and l are the same, the rule transforms matter, otherwise there is a communication and a transformation of substances across membranes.

Definition 6. (Multicompartimental Arithmetical MP System) *A Multicompartimental Arithmetical MP System of type* (n, m, k) *is a construct:*

$$M = (S, R, H, \Phi, \mathcal{L}, \mathcal{C}, \mathcal{K})$$

where:

S, H, Φ *are as in Definition 4;*
$\mathcal{L}$ *is set of membrane labels that are identified with the membranes their-selves.*
$\mathcal{C}$ *is a membrane structure describing the initial configuration.*
R *is a set of localized reactions associated to membranes* $k, l \in \mathcal{L}$ *with the forms:*

meta $[_k\, \alpha \rightarrow [_k\, \beta \ :: \varphi$
the classical MP reaction rewritten in the multicompartimental context: in membrane k *at step* i, $\widehat{\varphi}(X[i])$ *units of multiset* α *are transformed into* $\widehat{\varphi}(X[i])$ *units of multiset* β.

in $\alpha\, [_k \rightarrow [_k\, \beta \ :: \varphi$
if α *is in the immediate outside region of membrane* k, $\widehat{\varphi}(X[i])$ *units of* α *are introduced in* k *and transformed into* $\widehat{\varphi}(X[i])$ *units of* β.

out $[_k\, \alpha \rightarrow \beta\, [_k \ :: \varphi$
if α *is contained in the membrane* k, *the reaction expels* $\widehat{\varphi}(X[i])$ *units of* α *from* k *transforming it into* $\widehat{\varphi}(X[i])$ *units of* β.

trans $[_k\, \alpha \rightarrow [_l\, \beta \ :: \varphi$
if α *is contained in the membrane* k *and in the immediate outside region of* k *there is the sibling membrane* l, *the reaction transforms* $\widehat{\varphi}(X[i])$ *units of* α *into* $\widehat{\varphi}(X[i])$ *units of* β *transferring it from* k *to* l.

extra-in $[_k \rightarrow [_k\, \alpha \ :: \varphi$
the reaction introduces $\widehat{\varphi}(X[i])$ *units of* α *into membrane* k *from the environment external to system* M.

extra-out $[_k\, \alpha \rightarrow [_k \ :: \varphi$
the reaction expels $\widehat{\varphi}(X[i])$ *units of* α *from membrane* k *into the environment external to system* M.

$\mathcal{K}$ *is a labelling of substances and membranes marking*
- *as input* In *and output* Out *substances or membranes:* In, Out $\subseteq S \times \mathcal{L}$
- *as* Start *and* Halt *only substances:* Start, Halt $\subseteq S$.

The dynamics of M *is written according to (2) following the semantics of the rules in* R.

The multicompartimental Arithmetical MP System defines a membrane model with flux regulators in which the model of calculus is deterministic and therefore effective for the implementations.

Example 1. (Abacus) Let's define a 3-digit abacus with carry in base 10. Two input numbers $N = 728$ and $M = 322$ are encoded, using the substance u inside the input membranes as $[7u]_{N_1}[2u]_{N_2}[8u]_{N_3}$ and $[3u]_{M_1}[2u]_{M_2}[2u]_{M_3}$. An insight is given in Fig. 5, where substances and membranes are denoted by circles and rounded boxes respectively. Substance u (not shown for simplicity) from input membranes is summed by transferring it digit-by-digit into the output membranes. The output membrane R_1 encodes the digit of thousands possibly

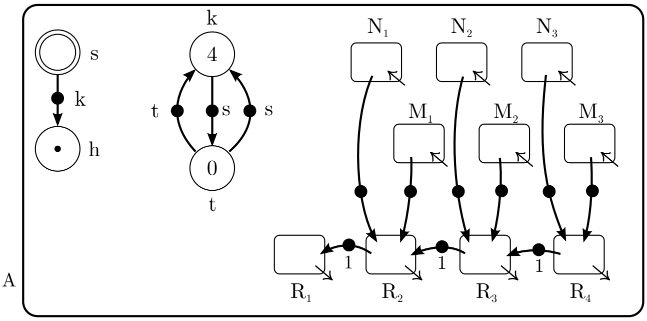


Fig. 5. A 3-digit abacus with carry

yielded by the carry rules, each trying to consume 10 units from its right region to produce 1 carry unit on its left region.

The computation, including the carry, may take up to $k = 4$ steps counted by the countdown network as follows. When s becomes 1, the reaction $r_{10} : k \rightarrow t$ occurs leaving $k = 3$ and $t = 1$. At each next step k is decreased, while the rules consuming t are blocked because they want to globally consume $t+s = t+1$ units of t. When k becomes 1 the rule $r_{13} : s \rightarrow h$ unblocks producing the HALT substance h and decreasing the value of s to 0. Now $r_{11} : t \rightarrow k$ can refill k making the counter network re-usable. The overall efficiency, as in digital computers, has an important limitation in the carry mechanism unless some additional optimizations are implemented. Notice that the reaction arrows can transform matter either in the same region or also communicate substances across membranes. The MP grammar in standard boundary notation is given by:

$$\begin{array}{rl} \textsc{In} = \{N_1, N_2, N_3, M_1, M_2, M_2\} \subseteq \mathcal{L} \\ \textsc{Out} = \{R_1, R_2, R_3, r_4\} \subseteq \mathcal{L} \\ \textsc{Start} = s \in S \\ \textsc{Halt} = h \in S \end{array} \qquad \begin{array}{ll} \overbrace{\hspace{8em}}^{R} & \overbrace{\hspace{3em}}^{\Phi} \\ r_1 : [_{N_1}\, u \rightarrow [_{R_2}\, u & \varphi_1 = u \\ r_2 : [_{M_1}\, u \rightarrow [_{R_2}\, u & \varphi_2 = u \\ r_3 : [_{N_2}\, u \rightarrow [_{R_3}\, u & \varphi_3 = u \\ r_4 : [_{M_2}\, u \rightarrow [_{R_3}\, u & \varphi_5 = u \\ r_5 : [_{N_3}\, u \rightarrow [_{R_4}\, u & \varphi_4 = u \\ r_6 : [_{M_3}\, u \rightarrow [_{R_4}\, u & \varphi_6 = u \\ & \\ r_7 : [_{R_4}\, 10u \rightarrow [_{R_3}\, u & \varphi_7 = 1 \\ r_8 : [_{R_3}\, 10u \rightarrow [_{R_2}\, u & \varphi_8 = 1 \\ r_9 : [_{R_2}\, 10u \rightarrow [_{R_1}\, u & \varphi_9 = 1 \\ & \\ r_{10} : [_A\, k \rightarrow [_A\, t & \varphi_{10} = s \\ r_{11} : [_A\, t \rightarrow [_A\, k & \varphi_{11} = t \\ r_{12} : [_A\, t \rightarrow [_A\, k & \varphi_{12} = s \\ r_{13} : [_A\, s \rightarrow [_A\, h & \varphi_{13} = k \end{array}$$

With the initial configuration
$\mathcal{C} = [_A\ 4K[7u]_{N_1}[2u]_{N_2}[8u]_{N_3}[3u]_{M_1}[2u]_{M_2}[2u]_{M_3}[]_{R_1}[]_{R_2}[]_{R_3}[]_{R_4}\ ]$.

4 Conclusions and a Glimpse on Lamp

Arithmetical MP systems are introduced as a computational model based on MP systems, a robust method for biological modeling [9]. The deterministic model of calculus, with data and processes localized in a membrane structure, come with the Lamp applicative framework.

Lamp is a new metabolic framework implementing a programming language and a vast repertoire of mechanisms for assembling MP Systems. Lamp is introduced online[1] giving the syntax in Extended Backus–Naur Form [11,4] along with semantics for all language-level mechanisms. The sub-routine mechanism is implemented with the *regulatory composition*, allowing to regulate the reaction rates by using the computational output of an adjacent system.

Deeper investigation is needed on algorithmic and modelling aspects as well as on the inter-operability with non-deterministic models and enhancements to the framework.

References

1. Bernardini, F., Manca, V.: Dynamical aspects of P systems. Biosystems 70(2), 85–93 (2003); Cell Computing
2. Bianco, L., Manca, V.: Encoding-decoding transitional systems for classes of P systems. In: Freund, R., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2005. LNCS, vol. 3850, pp. 134–143. Springer, Heidelberg (2006)
3. Díaz-Pernil, D., Pérez-Hurtado, I., Pérez-Jiménez, M.J., Riscos-Núñez, A.: A P-lingua programming environment for membrane computing. In: Corne, D.W., Frisco, P., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2008. LNCS, vol. 5391, pp. 187–203. Springer, Heidelberg (2009)
4. ISO/IEC. 14977: 1996(E) – Syntactic Metalanguage – Extended BNF. Technical report, International Organization for Standardization, Geneva, Switzerland (1996)
5. García-Quismondo, M., Gutiérrez-Escudero, R., Pérez-Hurtado, I., Pérez-Jiménez, M.J., Riscos-Núñez, A.: An overview of P-lingua 2.0. In: Păun, G., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rozenberg, G., Salomaa, A. (eds.) WMC 2009. LNCS, vol. 5957, pp. 264–288. Springer, Heidelberg (2010)
6. Manca, V.: Fundamentals of Metabolic P Systems. In: Handbook of Membrane Computing (2009)
7. Manca, V.: Metabolic P systems. Scholarpedia 5(3), 9273 (2010)
8. Manca, V., Bianco, L., Fontana, F.: Evolution and Oscillation in P Systems: Applications to Biological Phenomena. In: Mauri, G., Păun, G., Jesús Pérez-Jímenez, M., Rozenberg, G., Salomaa, A. (eds.) WMC 2004. LNCS, vol. 3365, pp. 63–84. Springer, Heidelberg (2005)
9. Manca, V., Marchetti, L.: Goldbeter's mitotic oscillator entirely modeled by MP systems. In: Proceedings of the Eleventh International Conference on Membrane Computing (CMC 2011), p. 277 (2010)
10. Manca, V., Marchetti, L.: Metabolic approximation of real periodical functions. Journal of Logic and Algebraic Programming 79(6), 363–373 (2010)

[1] Lamp, http://profs.sci.univr.it/~lombardo/papers/lamp2011.pdf

11. Naur, P. (ed.): Revised Report on the Algorithmic Language ALGOL 1960. Communications of the ACM, vol. 3, pp. 299–314 (May 1960)
12. Păun, G.: Computing with Membranes. Journal of Computer and System Sciences 61(1), 108–143 (2000)
13. Păun, G.: Membrane computing. In: Lingas, A., Nilsson, B.J. (eds.) FCT 2003. LNCS, vol. 2751, pp. 284–295. Springer, Heidelberg (2003)

Simulating Accepting Networks of Evolutionary Processors with Filtered Connections by Accepting Evolutionary P Systems

(Extended Abstract)

Juan Castellanos[1], Victor Mitrana[2], Eugenio Santos[2], and José M. Sempere[3]

[1] Department of Artificial Intelligence, Faculty of Informatics
Polytechnic University of Madrid, 28660 Boadilla del Monte, Madrid, Spain
jcastellanos@fi.upm.es

[2] Department of Organization and Structure of Information,
University School of Informatics,
Polytechnic University of Madrid, Crta. de Valencia km. 7 - 28031 Madrid, Spain
victor.mitrana@upm.es, esantos@eui.upm.es

[3] Department of Information Systems and Computation,
Polytechnic University of Valencia
Camino de Vera, s/n, 46022 Valencia, Spain
jsempere@dsic.upv.es

Abstract. In this work, we propose a variant of P system based on the rewriting of string-objects by means of evolutionary rules. The membrane structure of such a P system seems to be a very natural tool for simulating the filters in accepting networks of evolutionary processors with filtered connections. We discuss an informal construction supporting this simulation. A detailed proof is to be considered in an extended version of this work.

Keywords: Network of Evolutionary Processors with Filtered Connections, Evolutionary P system.

1 Introduction

A rather informal idea of what a network of evolutionary processor is consists of a virtual (complete) graph in which each node hosts a very simple processor called an evolutionary processor. This is a pretty common architecture for parallel and distributed symbolic processing, related to the Connection Machine [5] as well as the Logic Flow paradigm [3]. By an evolutionary processor we mean a processor which is able to perform very simple operations, namely point mutations in a DNA sequence (insertion, deletion or substitution of a pair of nucleotides). More generally, each node may be viewed as a cell having genetic information encoded in DNA sequences which may evolve by local evolutionary events, that is point mutations.

Each node processor, which is specialized just for one of these evolutionary operations, acts on the local data and then local data becomes a mobile agent

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 295–302, 2011.

which can navigate in the network following a given protocol. Only that data which is able to pass a filtering process can be communicated. This filtering process may require to satisfy some conditions imposed by the sending processor, by the receiving processor or by both of them. All the nodes send simultaneously their data and the receiving nodes handle also simultaneously all the arriving messages, according to some strategies, see [4,5]. The filtering process that is to be considered here is regulated by some context conditions associated to the edges of the graph. This is the model introduced in [2] under the name of accepting network of evolutionary processors with filtered connections (ANEPFC). The reader interested in a survey of the main results regarding ANEPFCs is referred to [9].

P systems [13] were introduced as a computational model inspired by the information and biochemical product processing of living cells through the use of membrane communication. In most of the works about P systems, information is represented as multisets of symbol/objects which can interact and evolve according to predefined rules. Nevertheless, the use of strings to represent the information and the use of rules to transform strings instead of multiset objects have always been present in the literature of this scientific area. So, in his mostly referred book [13], Gh. Păun overviews the use of string rules in P systems. Different variants of string-based P systems have been proposed along the time. We can mention *rewriting P systems* [10], referred as membrane systems with *worm objects* [1] in the case of genomic operations, *insertion-deletion P systems* [7] and *splicing P systems* [12], among others. Observe that most of these models have been used for language generation [11]. In [6,8], the proposal of *hybrid P systems* introduces the use of contextual rules and Chomsky rules to achieve universality by generating all the recursively enumerable languages.

In this work, we propose the use of evolutionary rules of string rewriting in all regions of a P system. The idea is not new (see for instance [7]) but our approach has two different main goals. First, the P systems considered here define languages by an accepting process in contrast with the generating variants widely considered so far in the area of membrane computing. Second, we show that the membrane structure is indispensable for simulating the filtering process in an ANEPFC. This is also in contrast with very many constructions of P systems in which the membrane structure plays actually a very minor role, most of them being reduced to just one membrane. The main part of this note is the informal construction of an evolutionary P system simulating an ANEPFC with emphasis on the membrane hierarchies of each region associated with a filtered connection.

The structure of this work is as follows: In section 2 and 3 we recall the definition of ANEPFCs and evolutionary P systems, respectively. Then, we informally describe a construction of an evolutionary P system simulating a given ANEPFC. More precisely, we discuss the evolutionary rules in each region and the membrane structure associated with each filtered connection in the simulated network.

2 Accepting Networks of Evolutionary Processors with Filtered Connections

We start by summarizing the notions used throughout this work. An *alphabet* is a finite and nonempty set of symbols. The cardinality of a finite set A is written $card(A)$. Any finite sequence of symbols from an alphabet V is called *string* over V. The set of all strings over V is denoted by V^* and the empty string is denoted by ε. The length of a string x is denoted by $|x|$ while $alph(x)$ denotes the minimal alphabet W such that $x \in W^*$.

We say that a rule $a \to b$, with $a, b \in V \cup \{\varepsilon\}$ and $ab \neq \varepsilon$ is a *substitution rule* if both a and b are not ε; it is a *deletion rule* if $a \neq \varepsilon$ and $b = \varepsilon$; it is an *insertion rule* if $a = \varepsilon$ and $b \neq \varepsilon$. The set of all substitution, deletion, and insertion rules over an alphabet V are denoted by Sub_V, Del_V, and Ins_V, respectively.

Given a rule σ as above and a string $w \in V^*$, we define the following *actions* of σ on w:

- If $\sigma \equiv a \to b \in Sub_V$, then $\sigma^*(w) = \begin{cases} \{ubv : \exists u, v \in V^* \ (w = uav)\}, \\ \{w\}, \text{ otherwise} \end{cases}$
- If $\sigma \equiv a \to \varepsilon \in Del_V$, then $\sigma^*(w) = \begin{cases} \{uv : \exists u, v \in V^* \ (w = uav)\}, \\ \{w\}, \text{ otherwise} \end{cases}$

$$\sigma^r(w) = \begin{cases} \{u : \ w = ua\}, \\ \{w\}, \text{ otherwise} \end{cases} \qquad \sigma^l(w) = \begin{cases} \{v : \ w = av\}, \\ \{w\}, \text{ otherwise} \end{cases}$$

- If $\sigma \equiv \varepsilon \to a \in Ins_V$, then

$$\sigma^*(w) = \{uav : \exists u, v \in V^* \ (w = uv)\}, \ \sigma^r(w) = \{wa\}, \ \sigma^l(w) = \{aw\}.$$

$\alpha \in \{*, l, r\}$ expresses the way of applying a deletion or insertion rule to a string, namely at any position ($\alpha = *$), in the left ($\alpha = l$), or in the right ($\alpha = r$) end of the string, respectively. For every rule σ, action $\alpha \in \{*, l, r\}$, and $L \subseteq V^*$, we define the *α-action of σ on L* by $\sigma^\alpha(L) = \bigcup_{w \in L} \sigma^\alpha(w)$. Given a finite set of rules M, we define the *α-action of M* on the string w and the language L by:

$$M^\alpha(w) = \bigcup_{\sigma \in M} \sigma^\alpha(w) \ \text{ and } \ M^\alpha(L) = \bigcup_{w \in L} M^\alpha(w),$$

respectively. In what follows, we shall refer to the rewriting operations defined above as *evolutionary operations* since they may be viewed as linguistic formulations of local DNA mutations.

For two disjoint subsets P and F of an alphabet V and a string w over V, we define the predicates:

$$\begin{aligned} \varphi^{(s)}(w; P, F) &\equiv \ P \subseteq alph(w) && \wedge \ F \cap alph(w) = \emptyset \\ \varphi^{(w)}(w; P, F) &\equiv alph(w) \cap P \neq \emptyset && \wedge \ F \cap alph(w) = \emptyset. \end{aligned}$$

The construction of these predicates is based on *random-context conditions* defined by the two sets P (*permitting contexts/symbols*) and F (*forbidding contexts/symbols*). Informally, the first condition requires that all permitting symbols are present in w and no forbidding symbol is present in w, while the second one is a weaker variant of the first, requiring that at least one permitting symbol appears in w and no forbidding symbol is present in w. For every language $L \subseteq V^*$ and $\beta \in \{(s), (w)\}$, we define:

$$\varphi^{\beta}(L,P,F)=\{w\in L\mid \varphi^{\beta}(w;P,F)\}.$$

An *evolutionary processor over* V is a tuple (M,PI,FI,PO,FO), where:

• M is a set of substitution, deletion or insertion rules over the alphabet V. Formally: $(M\subseteq Sub_V)$ or $(M\subseteq Del_V)$ or $(M\subseteq Ins_V)$. The set M represents the set of evolutionary rules of the processor. As one can see, a processor is "specialized" in one evolutionary operation, only.
• $PI,FI\subseteq V$ are the *input* permitting/forbidding contexts of the processor, while $PO,FO\subseteq V$ are the *output* permitting/forbidding contexts of the processor. Informally, the permitting input/output contexts are the set of symbols that should be present in a string, when it enters/leaves the processor, while the forbidding contexts are the set of symbols that should not be present in a string in order to enter/leave the processor.

We denote the set of evolutionary processors over V by EP_V.

An *accepting network of evolutionary processors with filtered connections* (ANEPFC for short) is a 8-tuple $\Gamma=(V,U,G,\mathcal{R},\mathcal{N},\alpha,\beta,x_I,x_O)$, where:

◇ V and U are the input and network alphabet, respectively, $V\subseteq U$.
◇ $G=(X_G,E_G)$ is an undirected graph without loops with the set of vertices X_G and the set of edges E_G. G is called the *underlying graph* of the network.
◇ $\mathcal{R}:X_G\longrightarrow 2^{Sub_U}\cup 2^{Del_U}\cup 2^{Ins_U}$ is a mapping which associates with each node the set of evolutionary rules that can be applied in that node. Note that each node is associated only with one type of evolutionary rules, namely for every $x\in X_G$ either $\mathcal{R}(x)\subset Sub_U$ or $\mathcal{R}(x)\subset Del_U$ or $\mathcal{R}(x)\subset Ins_U$ holds.
◇ $\mathcal{N}:E_G\longrightarrow 2^U\times 2^U$ is a mapping which associates with each edge $e\in E_G$ the disjoint sets $\mathcal{N}(e)=(P_e,F_e)$, $P_e,F_e\subset U$.
◇ $\beta:E_G\longrightarrow\{s,w\}$ defines the *filter* type of an edge.

We say that $card(X_G)$ is the size of Γ.

A *configuration* of an ANEPFC Γ as above is a mapping $C:X_G\longrightarrow 2^{V^*}$ which associates a set of strings with every node of the graph. A configuration may be understood as the sets of strings which are present in any node at a given moment. A configuration can change either by an *evolutionary step* or by a *communication step.*

When changing by an evolutionary step each component $C(x)$ of the configuration C is changed in accordance with the set of evolutionary rules M_x associated with the node x and the way of applying these rules $\alpha(x)$. Formally, we say that the configuration C' is obtained in *one evolutionary step* from the configuration C, written as $C\Longrightarrow C'$, if and only if

$$C'(x)=M_x^{\alpha(x)}(C(x))\text{ for all }x\in X_G.$$

When changing by a communication step, each node processor $x\in X_G$ of an ANEPFC sends one copy of each string it has to every node processor y connected to x, provided they can pass the filter of the edge between x and y. It keeps no copy of these strings but receives all the strings sent by any node processor z connected with x providing that they can pass the filter of the edge between x and z.

Formally, we say that the configuration C' is obtained in *one communication step* from configuration C, written as $C \vdash C'$, iff

$$C'(x) = (C(x) \setminus (\bigcup_{\{x,y\} \in E_G} \varphi^{\beta(\{x,y\})}(C(x), \mathcal{N}(\{x,y\}))))$$
$$\cup(\bigcup_{\{x,y\} \in E_G} \varphi^{\beta(\{x,y\})}(C(y), \mathcal{N}(\{x,y\})))$$

for all $x \in X_G$. Note that a copy of a string remains in the sending node x only if it not able to pass the filter of any edge connected to x.

Let Γ be an ANEPFC, the computation of Γ on the input string $w \in V^*$ is a sequence of configurations $C_0^{(w)}, C_1^{(w)}, C_2^{(w)}, \ldots$, where $C_0^{(w)}$ is the initial configuration of Γ defined by $C_0^{(w)}(x_I) = \{w\}$ and $C_0^{(w)}(x) = \emptyset$ for all $x \in X_G$, $x \neq x_I$, $C_{2i}^{(w)} \Longrightarrow C_{2i+1}^{(w)}$ and $C_{2i+1}^{(w)} \vdash C_{2i+2}^{(w)}$, for all $i \geq 0$. By the previous definitions, each configuration $C_i^{(w)}$ is uniquely determined by the configuration $C_{i-1}^{(w)}$. A computation as above is said to be an *accepting computation* if there exists a configuration in which the set of strings existing in the output node x_O is non-empty. The *language accepted* by Γ is

$L(\Gamma) = \{w \in V^* \mid$ the computation of Γ on w is an accepting one$\}$.

We denote by $\mathcal{L}(ANEPFC)$ the class of languages accepted by ANEPFCs.

3 Accepting Evolutionary P Systems

We now informally describe the P system we are going to investigate. An *Accepting evolutionary P system* (AEPS for short) of degree m is a construct

$$\Pi = (V, U, \mu, (R_1, \rho_1), \cdots, (R_m, \rho_m)),$$

where:

- V is the input alphabet, $U \supseteq V$ is the working alphabet,
- μ is a membrane structure consisting of m membranes,
- R_i, $1 \leq i \leq m$ is a finite set of *evolutionary and/or dissolving rules* over U associated with the ith region and ρ_i is a partial order relation over R_i specifying the *priority* among the rules. An evolutionary rule is a 4-tuple (a, b, α, β) (or $a \to b_\alpha^\beta$) where $a, b \in U \cup \{\lambda\}$, $\alpha \in \{here, out, in\}$ and $\beta \in \{*, l, r\}$. A dissolving rule is 5-tuple $(a, b, \alpha, \beta, \delta)$ (or $a \to b_\alpha^\beta \delta$), where a, b, α, β have the same meaning as for evolutionary rules and $\delta \in U$ is the dissolving symbol.

The application of a rule $a \to b_\alpha^\beta$ in an arbitrary region of the system works as follows: if there exists a string w in that region, such that $w = u_1 a u_2$, then w is transformed into $u_1 b u_2$ (observe that β establishes the way of applying the evolutionary rule). Parameter α establishes where to send the new strings, namely they are sent to the outer region, to all immediate inner regions (a copy of each string is sent to all these regions), or remain in the same region, provided that β is *out*, *in*, or *here*. If a string is to be sent to an inner region that does not exist, then it remains where it is.

If the rule is a dissolving one ($a \to b_{\alpha}^{\beta}\delta$, the membrane of the region is dissolved after the rule application, provided that the membrane is different from the skin one.

The input string is initially stored in the outmost region. Then, in a fully parallel manner all the rules are applied to the strings existing in every region according to their priorities. The system halts whenever: (1) No rule can be applied, or (2) The system is reduced to only one region, namely the outmost one.

The language accepted by Π is denoted by $L(\Pi)$. A string is in $L(\Pi)$ if and only if it being initially stored in the outmost region reduces the system to only one region.

4 A Simulation of ANEPFCs by AEPSs

In this section we give just a very brief idea how an AEPS, say Π, can simulate an ANEPFC, say Γ. The membrane structure of Π consists of the skin region that includes and as many regions as connections between the processors of Γ. For every connection between processors i and j we have the regions R_{ij} and R_{ji} inside the skin membrane. Each region R_{ij} has different structure depending on the filter type. A symbol in the current string in Π indicates the fact that the corresponding string in Γ has just arrived in the node i. We discuss the simulation of one evolutionary step in the node i and the communication process of the words from the node i to the node j in Γ.

Let us suppose that the set of permitting symbols for the filter on the connection between processor i and j in Γ is defined by $P_{ij} = \{b_1, b_2, \cdots, b_k\}$. If this filter acts in the weak mode, then the membrane structure in the region R_{ij} is showed in the left part of the next figure, while if the filter acts in the strong mode, then the membrane structure is showed in the right part of the same figure.

Inside the inner regions in illustrated membrane structures we apply the same evolutionary rules as those associated with the node i. In addition, the new strings are moved through the membrane structure in order to check the filter conditions. Thus evolutionary rules having the highest priority check the presence of forbidden symbols. If such a symbol is present, then the string remains

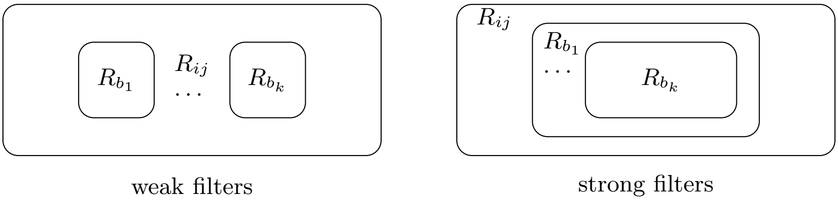


Fig. 1. Membrane structures for the filters

blocked in an inner region. If these rules cannot be applied, then other evolutionary rules check the presence of permitting symbols. As soon as one permitting symbol is present, the string is sent to the outer region. Clearly, some special symbols are used in order to manage the string movements. It is worth noting the important role of the membrane structure in each region R_{ij} by an analogy with electronic circuits: the membranes checking in the weak mode the presence of a permitting symbol form a sort of parallel circuit, while the membranes checking in the strong mode the presence of a permitting symbol form a sort of serial circuit.

When a string is going to enter a region of the form R_{ij} where i is the output node of the ANEPFC, then a new symbol is inserted; this symbol will dissolve in turn all the membranes.

5 Conclusions and Future Work

In this paper we have proposed a string accepting P system based on a generalization of the evolutionary rules considered for ANEPFCs. We have intuitively described a construction of an AEPS able to naturally simulate an ANEPFC. We consider that the simulation process is natural as the membrane structure is of a great importance in the simulation of the filtering process in ANEPFC. It is worth mentioning that this is in contrast with very many constructions of P systems in which the membrane structure plays actually a very minor role, most of them being reduced to just one membrane.

A technical proof together with other computationally aspects of accepting evolutionary P systems are to be considered for an extension of this note.

References

1. Castellanos, J., Păun, G., Rodríguez-Patón, A.: P systems with worm-objects. In: Proc. of the Seventh International Symposium on String Processing Information Retrieval (SPIRE 2000), pp. 64–74. IEEE Computer Society, Los Alamitos (2000)
2. Drăgoi, C., Manea, F., Mitrana, V.: Accepting networks of evolutionary processors with filtered connections. Journal of Universal Computer Science 13, 1598–1614 (2007)
3. Errico, L., Jesshope, C.: Towards a new architecture for symbolic processing. In: Artificial Intelligence and Information-Control Systems of Robots 1994, pp. 31–40 (1994)
4. Fahlman, S.E., Hinton, G.E., Seijnowski, T.J.: Massively parallel architectures for AI: NETL, THISTLE and Boltzmann machines. In: Proc. of the National Conference on Artificial Intelligence, pp. 109–113 (1983)
5. Hillis, W.D.: The Connection Machine. MIT Press, Cambridge (1979)
6. Krishna, S.R., Lakshmanan, K., Rama, R.: Hybrid P systems. Romanian Journal of Information Science and Technology 4(1-2), 111–123 (2001)
7. Krishna, S.R., Rama, R.: Insertion-deletion P systems. In: Jonoska, N., Seeman, N.C. (eds.) DNA 2001. LNCS, vol. 2340, pp. 360–370. Springer, Heidelberg (2002)
8. Madhu, M., Krithivasan, K.: A note on hybrid P systems. Grammars 5(3), 239–244 (2002)

9. Manea, F., Martín-Vide, C., Mitrana, V.: Accepting networks of evolutionary word and picture processors: A survey. In: Scientific Applications of Language Methods. Mathematics, Computing, Language, and Life: Frontiers in Mathematical Linguistics and Language Theory, vol. 2, pp. 523–560. World Scientific, Singapore (2010)
10. Martín-Vide, C., Păun, G.: String-objects in P systems. In: Proc. of Algebraic Systems, Formal Languages and Computations Workshop, RIMS Kokyuroku, Kyoto Univ., pp. 161–169 (2000)
11. Martín-Vide, C., Păun, G.: Language generating by means of membrane systems. Bulletin of the EATCS (75), 199–218 (October 2001)
12. Păun, A., Păun, M.: On membrane computing based on splicing. In: Words, Sequences, Languages, pp. 409–422. Kluwer Academic Publishers, Dordrecht (2000)
13. Păun, G.: Membrane Computing. An Introduction. Springer, Heidelberg (2002)

Towards the Automatic Programming of NEPs

Emilio del Rosal, Marina de la Cruz, and Alfonso Ortega de la Puente*

Departamento de Ingeniería Informática, Escuela Politécnica Superior,
Universidad Autónoma de Madrid
{emilio.delrosal,marina.cruz,alfonso.ortega}@uam.es

Abstract. This paper shows the platform with which we implement a general methodology to automatically design NEPs to solve specific problems. We use CGE/AGE (a new genetic programming algorithm) and jNEP (a Java NEP simulator), two applications we have previously developed. This work is just a proof of viability. We are interested on linking all the modules and generating the initial population. Building this platform is relevant, because our methodology includes several non trivial steps, such as designing a grammar, and implementing and using a simulator. For this first proof we have choosen a well known problem that other authors have solved by means of NEPs.

1 Motivation

The task of automatically writing programs can be seen as a search problem: finding the best in a set of candidate programs automatically generated. Any general search technique can be used to solve this problem.

Conventional personal computers are based on the well known von Neumann architecture, that can be considered as an implementation of the Turing machine. A great effort is being devoted to the design of new abstract computing devices, which can be seen as alternative architectures to design new families of computers. Some of them, inspired in the way used by Nature to solve difficult tasks efficiently, are called *natural or unconventional computers*. Some of the natural phenomena inspiring these devices are: the role of membranes in the behaviour of cells, the structure of genetic information, biological networks that perform natural processes in parallel, and the way in which species evolve.

Any computer scientist has a clear idea about how to program *conventional* (von Neumann) computers by means of different high level programming languages and their corresponding compilers, which translate programs into machine code. On the other hand, imagining how to program unconventional computers is quite difficult.

This paper shows a new platform using evolutionary computation for the automatic programming of a family of natural computers (Networks of Evolutionary Processors or NEPs).

* This work was partially supported by the R&D program of the Community of Madrid (S2009/TIC-1650, project "e-Madrid"). The authors thank Dr. Manuel Alfonseca for his help to prepare this document.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 303–312, 2011.

Some of the authors of this paper have proposed new evolutionary automatic programming algorithms (Attribute Grammar Evolution, AGE, [5] and Christiansen Grammar Evolution, CGE, [11]) as powerful tools to design complex systems to solve specific tasks.

Both techniques, AGE and CGE, wholly describe the candidate solutions, both syntactically and semantically, by means, respectively, of attribute and Christiansen grammars; thus improving the performance of other approaches, because they reduce the search space by excluding non-promising individuals with syntactic or semantic errors.

NEPs are abstract devices with a complex structure, because some of their components depend on others. This dependence makes it difficult to use genetic techniques to search NEPs because, in this circumstance, genetic operators usually produce a great number of incorrect individuals (either syntactically or semantically).

This paper shows the steps we have taken to implement a real platform to automatically design (or program) NEPs by means of our genetic algorithms (AGE/CGE). We have chosen a well known family of NEPs, able to solve a very simple problem [4]: the application of context free rules by classic NEPs. It is worth noticing that context free rules are not allowed in the classic family of NEPs. In this family, it is just allowed to replace a symbol by a single symbol (rather than a string of symbols, as in context free grammars). We are currently interested in developing the complete platform and have designed a static (non adaptable) simple context free grammar (without attributes) that could be used as the kernel for further Christiansen or attribute versions. This grammar generates a family of NEPs that includes that of [4]. To reduce the size of the search space, we have fixed all the components of the NEPs except the sets of rules and filters. We also have used an empty fitness function, because our goal is just to generate a valid initial population. In the future, we will include our simulator (jNEP [6]) in the fitness function used to evolve the initial individuals.

Our final goal is to test our techniques for the automatic design of NEPs to solve given tasks. We hope that our experiments may result in the proposal of a methodology to automatically design NEPs.

Figure 1 graphically describes the different blocks which can be considered to propose a general way to program natural computers similar to NEPs to solve a given problem.

This method takes as inputs the following elements:

- The *target problem* to be solved.
- The *computing device* that will be used to solve the problem.

And it consists of the following modules:

- An *evolutionary engine*, used as an automatic programming algorithm. This engine has to handle candidate solutions with a complex structure. We propose using AGE or CGE.
- A *formal description of the computing device* being programmed.
- A *simulator for the computing device* that will be used to compute the fitness function.

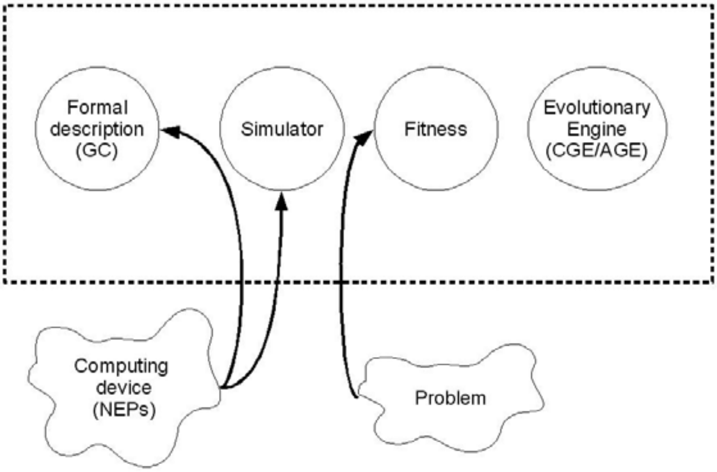


Fig. 1. Blocks of a general way to program natural computers

- The *fitness function*, which must fulfill two roles:
  1. Simulate the generated solution (in this case, a particular NEP).
  2. Measure how well the solution solves the target problem.

This paper shows the skeleton of our implementation of this methodology for NEPs. To generate a correct initial population is the first mandatory step to test its viability.

In the following sections, we will briefly introduce the main topics related to our research (AGE/CGE, NEPs). We will next describe our platform. Finally we will highlight our further research lines.

2 Introduction to AGE/CGE

Attribute grammars (AG) [7] are one of the tools used to completely describe high level programming languages (both their syntax and their semantics). Christiansen Grammars (CG) [3] are an adaptable extension to AG, that is, they are attribute grammars that modify themselves while they are used.

AGE [5] and CGE [11] are extensions to Grammatical Evolution [10]. Both techniques are automatic programming evolutionary algorithms independent of the target programming language, and include a standard representation of genotypes as strings of integers (codons), and a formal grammar (respectively attribute and Christiansen grammar) as inputs for the deterministic mapping of a genotype into a phenotype. This mapping minimizes the generation of syntactically and also semantically invalid phenotypes. Genetic operators act at the genotype level, while the fitness function is evaluated on the phenotypes.

Further details, deeper descriptions and examples can be found in [5] and [11].

3 Introduction to NEPs

Networks of evolutionary processors (NEPs) [1] are a new computing mechanism directly inspired in the behaviour of cell populations. Each cell contains its own genetic information (represented by a set of strings of symbols) that is changed by some *evolutive* transformations implemented as elemental operations on strings. Cells are interconnected and can exchange information (strings) with other cells.

A NEP can be defined as a graph whose nodes are processors which perform very simple operations on strings and send the resulting strings to other nodes. Every node has filters that block some strings from being sent and/or received.

NEPs: Definitions and Key Features. Following [1] we introduce the basic definition of NEPs.

Definition 1. *A Network of Evolutionary Processors of size n is a construct:*

$$\Gamma = (V, N_1, N_2, ..., N_n, G),$$

where:

- V is an alphabet.
- For each $1 \leq i \leq n$, $N_i = (M_i, A_i, PI_i, PO_i)$ is the i-th evolutionary node processor of the network. The parameters of every processor are:
  - M_i is a finite set of evolution rules of just one of the following forms:
    i. $a \to b$, where $a, b \in V$ (substitution rules),
    ii. $a \to \varepsilon$, where $a \in V$ (deletion rules),
    iii. $\varepsilon \to a$, where $a \in V$ (insertion rules).
  - A_i is a finite set of strings over V. The set A_i is the set of initial strings in the i-th node.
  - PI_i and PO_i are subsets of V^* respectively representing the input and the output filters. These filters are defined by the membership condition, namely a string $w \in V^*$ can pass the input filter (the output filter) if $w \in PI_i (w \in PO_i)$. In this paper we will use two kind of filters:
    - ∗ Those defined as two components (P, F) of Permitting and Forbidding contexts (a word w passes the filter if $(\alpha(w) \subseteq P) \wedge (F \cap \alpha(w) = \emptyset)$, where α represents the set of symbols of a given string).
    - ∗ Those defined as regular expressions r (a word w passes the filter if $w \in L(r)$, where $L(r)$ stands for the language defined by the regular expression r).
- $G = (\{N_1, N_2, \ldots, N_n\}, E)$ is an undirected graph called the underlying graph of the network. The edges of G, that is the elements of E, are given in the form of sets of two nodes. The complete graph with n vertices is denoted by K_n.

A configuration of a NEP is an n-tuple $C = (L_1, L_2, \ldots, L_n)$, with $L_i \subseteq V^*$ for all $1 \leq i \leq n$. It represents the sets of strings which are present in any node at a given moment.

A given configuration of a NEP can change either by an *evolutionary* step or by a *communicating* step. When changing by an evolutionary step, each component L_i of the configuration is changed in accordance with the evolutionary rules associated with the node i.

When changing by a communication step, each node processor N_i sends all the copies of the strings it has, able to pass its output filter, to all the node processors connected to N_i, and receives all the copies of the strings sent by any node processor connected with N_i, if they can pass its input filter.

4 Automatic Programming of NEPs

According to the general methodology previously outlined, we have decided to use a Java NEP simulator (jNEP[1] [6]). The grammar and the fitness function depend on this choice.

jNEP uses as inputs XML files describing the NEP being simulated. Our individuals will be valid XML descriptions of the NEPs[2]. This paper shows the context free grammar we are currently using to generate the initial population. It will be the kernel for the Christiansen grammar we will finally use in further experiments. AGE/CGE are able to include semantic constraints when generating the populations to ensure that only syntactically and semantically valid individuals belong to them. When we design AGE/CGE experiments, we have to tune also the *amount of semantic constraints* we add to the grammar. We have seen that *too much semantics* converts the genetic process into a random search. This is the reason why we have decided to remove these restrictions and use a context free grammar in our first proofs.

In this work we have not used all the options available for describing NEPs by means of jNEP XML files. jNEP accepts all the variants and constructs found in the literature. In order to reduce the huge search space defined by the full grammar, we have decided to force some structural and functioning characteristics of NEPs. In the future we will test the system with more general NEPs. The features of the NEPs generated are explained in the following section.

Our goal is to solve a moderately complex problem presented and solved in [4]. This paper shows how NEPs simulate the application of context free rules ($A \rightarrow \alpha, A \in V, \alpha \in V^*$ for alphabet V) in three steps. The first one (that is our goal) rotates the string where the rule is being applied until placing A in one of the string ends. In [4], this taks is performed by means of a NEP with 4 nodes sequentially connected.

The evolution of this NEP can be summarized as follows: let us call “s” the symbol being rotated. The first node of the NEP receives a word where the symbol “s” is at the end. This node substitutes “s” by an auxiliary copy (“s_a1”). Then, the new word is sent to the second node, where a new auxiliary symbol (“s_a2”) is added to the beginning of the word. The last two nodes remove “s_a1” and substitute “s_a2” by the original“s”. These nodes use filters that reject those

[1] The jNEP code is freely available in http://jnep.e-delrosal.net

[2] See the jNEP User Guide in http://jnep.e-delrosal.net

words without auxiliary symbols. At this point the rotation of “s” has finished. This cicle could be repeated as many times as needed until finding the symbol to which we want to apply the rule (“A”).

As we have explained before, we have decided to reduce the characteristics and functuonality of general NEPs. In this first proof we have used

- rules of all the kinds described in [8,1,9,2]
- filters based on *random context conditions*, that is, the four usual types of filters described in [8]
- the same graph structure as in [4]
- In addition, we have bounded the number of symbols of the alphabet (see the grammar below)

This is the context free grammar we have used (notice that symbols “[” and “]” are used to enclose non terminal symbols in the right hand side of the rules while the XML markers for tags “<” and “>” have to appears literally):

```
                  NEP ::= <?xml version=”1.0”?> <NEP nodes=“[nodes]”> [alphabetTag] [graphTag]
                          [processorsTag] [stoppingConditionsTag] </NEP>
                nodes ::= 5
          alphabetTag ::= <ALPHABET symbols=“a_b_c_o_p_q_r_s_t_u_v_w_x_y_z”/>
             graphTag ::= <GRAPH> <EDGE vertex1=“0” vertex2=“1”/> <EDGE vertex1=“1”vertex2=“2”/>
                          <EDGE vertex1=“2” vertex2=“3”/> <EDGE vertex1=“3” vertex2=“4”/> </GRAPH>
        processorsTag ::= <EVOLUTIONARY_PROCESSORS> [nodeTagInit] [nodeTag] [nodeTag] [node-
                          Tag] [nodeTag] </EVOLUTIONARY_PROCESSORS>
          nodeTagInit ::= <NODE initCond=”input”> [evolutionaryRulesTag] [nodeFiltersTag] </NODE>
              nodeTag ::= <NODE initCond=””> [evolutionaryRulesTag] [nodeFiltersTag] </NODE>
 evolutionaryRulesTag ::= <EVOLUTIONARY_RULES> [ruleTag] </EVOLUTIONARY_RULES>
              ruleTag ::= <RULE ruleType=“[ruleType]” actionType=“[actionType]” symbol=“[symbol]”
                          newSymbol=“[symbol]”/> [ruleTag]
              ruleTag ::= λ
             ruleType ::= insertion | deletion | substitution
           actionType ::= LEFT | RIGHT | ANY
       nodeFiltersTag ::= <FILTERS>[inputFilterTag] [outputFilterTag]</FILTERS>
       nodeFiltersTag ::= <FILTERS>[inputFilterTag]</FILTERS>
       nodeFiltersTag ::= <FILTERS>[outputFilterTag]</FILTERS>
       nodeFiltersTag ::= <FILTERS></FILTERS>
       inputFilterTag ::= <INPUT [filterSpec]/>
      outputFilterTag ::= <OUTPUT [filterSpec]/>
           filterSpec ::= type = “[filterType]” permittingContext = “[symbolList]” forbiddingContext
                          = “[symbolList]”
           filterType ::= 1 | 2 | 3 | 4
             wordList ::= [symbolList] [wordList] | λ
           symbolList ::= [auxList] | λ
              auxList ::= [symbol] | [symbol]_[auxList]
               symbol ::= a|b|c|o|p|q|r|s|t|u|v|w|x|y|z
stoppingConditionsTag ::= <STOPPING_CONDITION> <CONDITION type = “NonEmptyNodeStop-
                          pingCondition” nodeID = “[lastNodeID]”/> <CONDITION type = “Maximum-
                          StepsStoppingCondition” maximum = “20”/> </STOPPING_CONDITION>
```

4.1 Testing the Framework

We have introduced the previous grammar in our CGE engine and run a preliminary evolutionary search. The framework creates a lot of different valid NEPs as expected. One of these NEPs is shown below.

An example of the generated NEPs. This NEP fullfils the functional and structural constraints imposed by the grammar. It has 4 nodes, connected sequentially.

The first one contains the input word that should be rotated by the solution NEP. One of the stopping conditions stop the NEP when a word reaches the last node, finishing the rotation. Others stop it after a given number of steps is executed, or when nothing has changed in two consecutive configurations.

However, rules and filters are almost unrestricted. In this example, the first node adds the symbol “t” at the end of its words and deletes any appearance of the symbol “p”. There is no output filter in the first node and no input filter in the second one. So, all the words enter the second node. The second node has no rule. Nothing changes from this point and the NEP stops.

The XML file for this NEP is shown below:

```
<?xml version="1.0"?>
  <NEP nodes="5">
  <ALPHABET symbols="a_b_c_o_p_q_r_s_t_u_v_w_x_y_z"/>
  <GRAPH>
    <EDGE vertex1="0" vertex2="1"/>
    <EDGE vertex1="1" vertex2="2"/>
    <EDGE vertex1="2" vertex2="3"/>
    <EDGE vertex1="3" vertex2="4"/>
  </GRAPH>
  <EVOLUTIONARY_PROCESSORS>
    <NODE initCond="input">
      <EVOLUTIONARY_RULES> <RULE ruleType="substitution" actionType="ANY"
                            symbol="z" newSymbol="u"/>
    <RULE ruleType="deletion" actionType="LEFT" symbol="p" newSymbol="u"/>
    <RULE ruleType="insertion" actionType="RIGHT" symbol="t" newSymbol="v"/>
      </EVOLUTIONARY_RULES> <FILTERS> </FILTERS>
    </NODE>
    <NODE initCond="">
      <EVOLUTIONARY_RULES></EVOLUTIONARY_RULES>
      <FILTERS></FILTERS>
    </NODE>
    <NODE initCond="">
      <EVOLUTIONARY_RULES> </EVOLUTIONARY_RULES>
      <FILTERS> <OUTPUT type="1" permittingContext="c" forbiddingContext="y"/>
      </FILTERS>
    </NODE>
    <NODE initCond="">
      <EVOLUTIONARY_RULES> </EVOLUTIONARY_RULES>
      <FILTERS>
    <INPUT type="2" permittingContext="" forbiddingContext="" />
    <OUTPUT type="1" permittingContext="" forbiddingContext="a_w_t" />
      </FILTERS>
    </NODE>
    <NODE initCond="">
      <EVOLUTIONARY_RULES> </EVOLUTIONARY_RULES>
      <FILTERS>
    <INPUT type="4" permittingContext="" forbiddingContext="" />
    <OUTPUT type="1" permittingContext="" forbiddingContext="" />
```

```
      </FILTERS>
    </NODE>
  </EVOLUTIONARY_PROCESSORS>
  <STOPPING_CONDITION>
    <CONDITION type="NonEmptyNodeStoppingCondition" nodeID="4"/>
    <CONDITION type="MaximumStepsStoppingCondition" maximum="15"/>
  </STOPPING_CONDITION>
</NEP>
```

We also show the output that jNEP generates while simulating this NEP.

```
XML CONFIGURATION FILE LOADED AND PARSED SUCCESSFULLY...
GRAPH INFO PARSED SUCCESSFULLY...
STOPPING CONDITIONS INFO PARSED SUCCESSFULLY...
EVOLUTIONARY PROCESSORS INFO PARSED SUCCESSFULLY...
****  NEP INITIAL CONFIGURATION  ****
        --- Evolutionary Processor 0 ---
input
        --- Evolutionary Processor 1 ---
...
        --- Evolutionary Processor 4 ---

****  NEP CONFIGURATION - EVOLUTIONARY STEP - TOTAL STEPS: 1  ****
        --- Evolutionary Processor 0 ---
input_t input
        --- Evolutionary Processor 1 ---
...
        --- Evolutionary Processor 4 ---

****  NEP CONFIGURATION - COMMUNICATION STEP - TOTAL STEPS:  2  ****
        --- Evolutionary Processor 0 ---

        --- Evolutionary Processor 1 ---
input_t input
        --- Evolutionary Processor 2 ---
...

****  NEP CONFIGURATION - EVOLUTIONARY STEP - TOTAL STEPS: 3  ****
        --- Evolutionary Processor 0 ---

        --- Evolutionary Processor 1 ---

        --- Evolutionary Processor 2 ---

        --- Evolutionary Processor 3 ---

        --- Evolutionary Processor 4 ---

...
```

```
---------------------- NEP has stopped!!! ----------------------

Stopping condition found:
net.e_delrosal.jnep.stopping.NoChangesStoppingCondition

-----------------------------------------------------------------

We are glad you used jNEP
```

5 Conclusions and Futher Research Lines

We have implemented an initial context free grammar for a family of NEPs able to solve a given problem: the application of context free rules. This family of NEPs has been taken from the literature ([4]). The same graph structure has been used, while the number of symbols of the alphabet has been limited. To find the proper rules and filters, our grammar generates valid XML files to be input into jNEP. Different initial populations have been successfully generated.

In order to check if our general methodology is applicable to this problem, we plan, in the future:

- To design and implement a proper fitness function.
- To add additional semantic constraints to our grammar, to drive the search.
- To design experiments to find, by means of CGE/AGE, different solutions to this problem.
- To compare the solucions automatically designed and that of ([4])

References

1. Castellanos, J., Martin-Vide, C., Mitrana, V., Sempere, J.M.: Networks of evolutionary processors. Acta Informatica 39(6-7), 517–529 (2003)
2. Castellanos, J., Martín-Vide, C., Mitrana, V., Sempere, J.M.: Solving NP-complete problems with networks of evolutionary processors. In: Mira, J., Prieto, A.G. (eds.) IWANN 2001. LNCS, vol. 2084, p. 621. Springer, Heidelberg (2001)
3. Christiansen, H.: A survey of adaptable grammars. SIGPLAN Notices 25(11), 35–44 (1990)
4. Csuhaj-Varju, E., Martin-Vide, C., Mitrana, V.: Hybrid networks of evolutionary processors are computationally complete. Acta Informatica 41(4-5), 257–272 (2005)
5. de la Cruz Echeandía, M., de la Puente, A.O., Alfonseca, M.: Attribute grammar evolution. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2005. LNCS, vol. 3562, pp. 182–191. Springer, Heidelberg (2005)
6. del Rosal, E., Nuñez, R., Castañda, C., Ortega, A.: Simulating neps in a cluster with jnep. In: Proceedings of International Conference on Computers, Communications and Control, ICCCC 2008 (2008)
7. Knuth, D.E.: Semantics of Context-Free Languages. Mathematical Systems Theory 2(2), 127–145 (1968)

8. Manea, F., Martín-Vide, C., Mitrana, V.: Solving 3CNF-SAT and HPP in linear time using WWW. In: Margenstern, M. (ed.) MCU 2004. LNCS, vol. 3354, pp. 269–280. Springer, Heidelberg (2005)
9. Martin-Vide, C., Mitrana, V., Perez-Jimenez, M.J., Sancho-Caparrini, F.: Hybrid networks of evolutionary processors. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 401–412. Springer, Heidelberg (2003)
10. ONeill, M., Conor, R.: Grammatical Evolution, evolutionary automatic programming in an arbitrary language. Kluwer Academic Publishers, Dordrecht (2003)
11. Ortega, A., de la Cruz, M., Alfonseca, M.: Christiansen grammar evolution: Grammatical evolution with semantics. IEEE Transactions on Evolutionary Computation 11(1), 77–90 (2007)

Bio-inspired Grammatical Inference

Leonor Becerra-Bonache

Laboratoire Hubert Curien, UMR CNRS 5516
Université de Saint-Etienne, Jean Monnet
Rue du Professeur Benoit Lauras, 42000 Saint-Etienne, France
leonor.becerra@univ-st-etienne.fr

Abstract. The field of Grammatical Inference was originally motivated by the problem of natural language acquisition. However, the formal models proposed within this field have left aside this linguistic motivation. In this paper, we propose to improve models and techniques used in Grammatical Inference by using ideas coming from linguistic studies. In that way, we try to give a new bio-inspiration to this field.

1 Introduction

The problem of how children acquire their first language has attracted the attention of researchers for many years. The desire to better understand natural language acquisition has motivated research in formal models of language learning [27,26]. Such models are of great interest for several reasons. On one hand, these models can help us to answer several key questions about natural language learning. On the other hand, these formal models can provide an operational framework for practical applications of language learning; for example, language learning by machines.

Grammatical Inference (GI) is a subfield of Machine Learning that deals with the learning of formal languages. The initial theoretical foundations of GI were given by E.M. Gold [18], who tried to formalize the process of natural language acquisition. After Gold's seminal work, research in this field has been specially focused on obtaining formal results (e.g., formal descriptions of the languages to be learned, formal proofs that a concrete algorithm can efficiently learn according to some concrete denitions, etc) [13]. Several formal models of language learning have been proposed in this field [17], however, such models do not take into account some important aspects of natural language acquisition, and assume idealized conditions as compared to the conditions under which children learn language (as we will see in Section 2). Therefore, although GI studies were motivated by the problem of natural language acquisition, its mathematization has left natural approaches aside.

Since the study of formal models of language learning is of great interest to better understand natural language acquisition, it is important that such models are inspired by studies of natural language acquisition. In that way, models can be more realistic, and can better simulate the human processing and acquisition of language.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 313–322, 2011.

Based on all these ideas, we propose to use ideas coming from linguistic studies to improve the models and techniques used in GI. In that way, we propose to give a new bio-inspiration to the field of GI, and bring back Grammatical Inference to its origins.

2 Grammatical Inference Studies

Grammatical Inference deals with the learning of formal languages from a set of data. The basic framework can be considered as a game played between two players: a teacher and a learner. The teacher provides information to the learner, and the learner must identify the underlying language from that information [13]. Excellent surveys on this field can be found in [28,16,17].

As we can see, the process of learning a formal language has some similarities with the process of language acquisition. For example, instead of a teacher and a learner, we have an adult and a child. Moreover, a child learns a language from the data that he/she receives (a child with an English environment will learn to speak English, and the same child with a Japanese environment will learn to speak Japanese). Therefore, GI provides a *good theoretical framework* for investigating the process of language learning.

The relevance of formal results in GI to the question of how children acquire their native language has been well recognized [33,13].

> *Positive results can help us to understand how humans might learn languages by outlining the class of algorithms that might be used by humans, considered as computational systems at a suitable abstract level. Conversely, negative results might be helpful if they could demonstrate that no algorithms of a certain class could perform the task in this case we could know that the human child learns his language in some other way* [13, p. 26].

However, work in the field of GI has been specially focused on the mathematical aspects of the language learning problem, leaving aside the linguistic motivation that originated these studies. Next we review some of the main drawbacks of GI from a linguistic point of view.

2.1 Formal Models in Grammatical Inference

Several formal models of language learning have been proposed in the field of GI. The main ones are: *Identification in the limit* [18], *Query learning model* [1], and *PAC learning model* [32]. The main drawback of these models is that they do not take into account some relevant aspects of natural language learning.

In the model proposed by E.M. Gold, *Identification in the limit*, there is no limit on how long the learner can take to guess the correct language; from a linguistic point of view, efficiency is important, since children are able to learn the language in an efficient way. Moreover, the learner passively receives strings of the language (but natural language learning is more than that, children also interact with their environment) and hypothesizes complete grammars instantaneously (this assumption is unrealistic).

The *Query learning model* proposed by D. Angluin has also some controversial aspects from a linguistic point of view; for example, the learner is able to ask the teacher if his hypothesis is correct (such a query will never be produced in a real situation; a child would never ask the adult if his grammar is the correct one), and the learner learns exactly the target language (this is not realistic, since everybody has imperfections in their linguistic competence).

In the *PAC learning model* proposed by L. Valiant, the examples provided to the learner have the same distribution throughout the process; this requirement is too strong for practical applications.

Therefore, none of these models perfectly account for natural language acquisition. Research in GI has been focused on the mathematical aspects of the formal models proposed, without exploiting their linguistic relevance. A longer discussion about these models can be found in [5].

2.2 Language Learning Problem

The problem of language learning concerns both the acquisition of the *syntax* (i.e., rules for generating and recognizing correct sentences in the language) and the *semantics* (i.e., the underlying meaning of each sentence) of a target language [26]. However, GI studies has been focused only on learning the syntax.

Semantics not only is one component of language learning, but also seems to play an important role in the first stages of children's language acquisition (as we will see in the next section). Therefore, it is also of great interest to study this component. Unfortunately, all these considerations have not been taken into account in GI studies; the learning problem has been reduced to syntax learning, and all semantic information has been omitted from their works.

GI algorithms are based on the availability of different types of information: positive examples, negative examples, the presence of a teacher able to answer queries, etc. However, *what kind of data is available to children?* Ideally, to better understand the process of natural language acquisition and to correctly simulate it, we should provide to our algorithm the same kind of examples that are available to children. However, some of the data used by GI algorithms are controversial from a linguistic point of view. We will discuss some linguistic studies that try to answer this question in the next section.

In order to make the problem of language learning well defined, it is also necessary to choose an appropriate class of grammars. The classes of *regular* and *context-free* grammars are often used in GI to model the target grammar. These two classes constitutes the first two levels of the Chomsky hierarchy. Thus, the following question arises: do they have enough expressive power to describe natural languages? From a linguistic point of view, it is of great interest to study classes of grammars that are able to generate the most relevant constructions that appear in natural languages. However, it seems not to be the case of regular and context-free grammars. We will discuss the limitations of the Chomsky hierarchy in the next section.

3 Linguistic Studies

The question of how children acquire their native language has been traditionally addressed by linguists. Their approach is specially focused on making experiments with children that are learning their native language. In that way, they try to collect data about the process of natural language acquisition (e.g., first sentences produced by children, errors made, etc.). Their final goal is to investigate the mental processes that occur during children's language acquisition, and try to describe this process.

There are different types of experiments. Depending on the way in which the data is collected, we have a *naturalistic approach* (i.e., samples of child language is collected or recorded in a comfortable environment. Data is collected regularly) or a *experimental approach* (i.e., the researcher proposes a work hypothesis and design specific tasks that have to be performed by the child to use specific language structures. A statistical analysis of the data is done at the end). Depending on the number of children used to do the experiments, we have *longitudinal studies* (i.e., experiments are focused just on one child, and they are done over a long period of time. Such approach is often combined with the naturalistic approach) or *transversal studies* (experiments are made with a group of children of different ages. Such approach is often combined with an experimental approach). And finally, depending on the kind of tasks performed by the child, we have experiments based on comprehension, production or imitation.

The CHILDES database (Child Language Data Exchange system) provides a large amount of useful data for linguistic studies of children's language acquisition. In this database we can find transcript and media data collected from conversations between children and adults. It has content in over 20 languages from 130 different corpora, all of which are available in http://childes.psy.cmu.edu/

Studies carried out in the field of Linguistics have helped to better understand some aspects of natural language acquisition [14]. However, there is still a lot of questions that do not have a clear answer; for example, what factors really have an effect in the process of children's language acquisition. Therefore, despite all investigations conducted so far, it has not been possible yet to understand all the rules, strategies, and other processes that underlie children's language acquisition.

Next we review some of the works and results obtained in this field, that are relevant for GI studies.

3.1 Data Available to Children

The question of what kind of data is available to children during the learning process is still a subject of discussion in Linguistics. It is generally accepted that children receive sentences that are grammatically correct, that is, *positive data*. However, the availability of another kind of data, usually called *negative data*, is a matter of controversy.

There are three different proposals to this question. The first proposal is that children do not receive negative data and they must rely on innate information to acquire their native language. This proposal is based on the Chomsky's *poverty of stimulus* argument: there are principles of the grammar that cannot be learnt from only positive data, and since children do not receive negative data (i.e., evidence about what is not grammatical), one can conclude that the innate linguistic capacity is what provides the additional knowledge that is necessary for language learning. Moreover, work presented by Brown and Hanlon in [11], has been used as an argument to support the unavailability of negative data to children. Concretely, they analyzed adult approval and disapproval of child utterances (for example, adult's answers as "That's right", "Correct", "That's wrong", "No"), and they found no relation between this type of answers and the grammaticality of the sentences produced by the children. However, this approach raises several questions: Should only explicit disapproval count as negative evidence? Could adults correct children in a different way?

The second proposal is that children receive negative data in the form of *different reply-types* given in response to grammatical versus ungrammatical child utterances. Hirsh-Pasek et al. [19] and Morgan and Travis [24] studied this type of negative evidence and proposed that parents respond to ungrammatical child utterances by using different types of answers from those they use when responding to grammatical utterances. Under this view, the reply-type would indicate to the child whether an utterance was grammatically correct or not. The problem of this second approach is that it does not take into account whether the adult's replies contain corrective information [12]. Moreover, under this approach, children would learn what utterances are correct only after complex statistical comparisons [23].

The third proposal is that children receive negative evidence in the form of *reformulations*, and they not only can detect them, but also they can make use of that information. Reformulations are sentences adults use in checking up on what their children intended to say (for example, a child says "milk milk" and the father answers "you want milk"?). Chouinard and Clark [12] proposed this new view of negative evidence. The main properties of this kind of corrections are the following: i) Adult's correction preserves the same meaning of the child; ii) Adult uses the correction to keep the conversation on track (adult reformulates the sentence just to make sure that he has understood the child's intentions); iii) Child's utterance and adult's correction have the same meaning, but different form.

It is worth noting that reformulations are often provided to children during the early stages of children's language acquisition. Moreover, semantics seems to play an important role in the first stages of children's language acquisition, concretely in the stage known as the two-word stage, in which children go through the production of one word to the combination of two elements [29,30].

3.2 Location of Natural Languages in the Chomsky Hierarchy

The question of where natural languages are located in the Chomsky hierarchy has been a subject of debate for a long time. This question was posed by Chomsky in the 50's. The debate was focused on whether natural language are context-free or not.

There were many attempts to prove the non-context-freeness of natural languages, but we have to wait until the late 80's to find solid arguments that support this idea. By that time, Bresnan et al. [10], Culy [15] and Shieber [31] presented some clear examples of natural language structures that cannot be described using a context-free grammar. Such examples were found in three different natural languages: Dutch, Bambara and Swiss-German. The kind of structures founded in these languages are: multiple agreements, crossed agreements and duplication. Therefore, after 20 years, linguists seemed to agree that context-free languages do not have enough expressiveness to describe the main context-sensitive constructions found in natural languages.

The discovery of non-context-free structures in natural languages aroused out the study and development of grammatical formalisms with more generative power than context-free. Since context-sensitive seems not to be the good solution (they are too powerful and computationally too complex), researchers tried to find another formal grammar more adequate to model natural language structures. The idea of generating context-free and non-context-free structures, keeping under control the generative power, has led to the notion of *Mildly Context-Sensitive* (MCS) grammars [20].

There exist very well known mechanisms to fabricate MCS families, for example, tree adjoining grammars, head grammars, combinatory categorial grammars. All of them occupy a concentric position in the Chomsky hierarchy, between context-free and context-sensitive. However, is it necessary that such formalism generates all context-free languages?

As some authors point out [21,22], natural languages could occupy an orthogonal position in the Chomsky hierarchy, that is, they contain some regular languages and some context-free languages, but they are included in context-sensitive. In fact, we can find natural language constructions that are neither regular nor context-free, and also some regular or context-free constructions that do not appear naturally in sentences.

4 Bio-inspired Grammatical Inference

In Section 2 we have discussed some of the main drawbacks of GI from a linguistic point of view. Concretely, we have seen that the most important models studied in GI do not take into account some important aspects of children's language acquisition and, consequently, they are quite unrealistic. The data used by the GI algorithms are also controversial from a linguistic point of view. Moreover, we have pointed out that works in GI tend to omit the semantic information and reduce the learning problem to syntax learning. And finally, we have seen that research in GI have been focused on learning regular and context-free languages, which constitutes the classes with less generative power of the Chomsky hierarchy.

In Section 3 we have discussed some of the works and results obtained in Linguistics, concerning natural language acquisition. Concretely, we have seen that there are three different proposals about the type of data that is available to the child. We have also pointed out the relevance of semantics in the first stages of children's language acquisition. And finally, we have seen that the Chomsky hierarchy has some limitations, specially when we try to locate natural languages in this hierarchy; since regular and context-free grammars seems not to be very adequate to model natural language syntax, linguists have tried to find other grammatical formalisms that have interesting linguistic and computational properties, such as Mildly Context-Sensitive.

Taking into account all these ideas, we propose to use linguistic studies to improve models and techniques used in GI. Thanks to ideas coming from linguistic studies on natural language acquisition, models in GI could be more realistic; these models could take into account more aspects about children's language acquisition. Moreover, such ideas could also improve the results obtained in the field of GI. In that way, we would use a *bio-inspired* model for language learning.

First of all, we propose that GI algorithms take into account not only *positive data*, but also *corrections* during the learning process. We consider that the most convincing proposal to the question of what kind of data is available to children, is the one proposed by Chouinard and Clark [12]. To consider that only explicit disapproval counts as negative data is not realistic. As Chouinard and Clark showed, adults correct children in a very different way, taking into account the meaning that the child intends to express. Moreover, a very large number of examples of such kind of meaning-preserving corrections can be found in real conversations between children and adults (for example, in CHILDES database). The second proposal is also unconvincing, since corrective and non-corrective replies are mixed in their analysis, and hence, learning from "reply-types" would require that children do complex statistical comparisons in order to learn which sentences are correct. Therefore, as Chouinard and Clark proposed in [12], we consider that meaning-preserving corrections are available to children, and they can help them to learn some aspects of natural language syntax.

In order to see the effect of corrections in language leaning, we propose to incorporate the idea of corrections to the studies of GI. We have already done some work in this direction. Our first approach has consisted on considering only *syntactic corrections* based on proximity between strings. Since this idea was totally new in GI, we started by learning deterministic finite automata [8], in the framework of query learning (i.e., the learner is able to ask queries to the teacher, and the teacher has to answer correctly to these questions). Later, these results were extended to learn other classes with interesting properties, such as *balls of strings* (which are defined by using the edit distance) [7]. In both cases, when the learner asks for a string that does not belong to the target language, the teacher returns a correction (in the first case, such correction is based on the shortest extension of the queried string, and in the second case, such correction is based on the edit distance). In both cases, we could show that results can be improved thanks to corrections. Therefore, such works show that new challenging results can be obtained in the field of GI if corrections are taken into account in the learning process.

Our second approach is based on *semantic corrections*. Corrections have a semantic component that has not been taken into account in previous works. Hence, we have proposed a new computational model of language learning that takes into account semantics. This model is bio-inspired by studies on children's language acquisition, and more concretely, by the results obtained by Chouinard and Clark in [12]. Our final goal has been to find a formal model that gives an account of *meaning preserving corrections*, and in which we can address the following questions: What are the effects of corrections on learning syntax? Can corrections facilitate the language learning process? Can semantic information simplify the problem of learning formal languages? It is worth noting that this has been the first attempt to incorporate a robust notion of semantics in the field of GI. Such a model has allowed us to investigate aspects of the roles of semantics and corrections in the process of learning to understand and speak a natural language. Our main results can be found in [4,3,2].

Taking into account that GI studies have been focused on learning regular and context-free languages, but, as linguistic studies suggest, these classes have limited expressive power to describe natural language syntax, we propose that GI studies focus on other classes such as *Mildly context-sensitive*. Moreover, we also support the idea that natural languages occupy an orthogonal position in the Chomsky hierarchy. Therefore, we propose to study formalisms that are able to generate MCS languages (i.e., they generate multiple agreement, crossed agreement and duplication structures, and they are computational feasible), and that occupy an orthogonal position in the Chomsky hierarchy (i.e., they contain some regular, some context-free, but they are included in context-sensitive). We have also done some works in this direction. We studied a mechanism that have such interesting properties, called *Simple External Contextual*. Our main results can be found in [9,25,6].

5 Conclusions

The field of GI provides a good theoretical framework for investigating the process of natural language acquisition. However, studies in this field have been focused on the mathematical aspects of the formal models proposed, without exploiting their linguistic relevance. Therefore, the linguistic motivation that originated GI studies has been left aside.

In this paper, we have discussed some linguistic studies on children's language acquisition and we have proposed to use them in order to improve models and techniques used in GI. Concretely, we have proposed that GI studies take into account *corrections* and *semantics* during the learning process, and they focus on classes of languages that are MCS and occupy an orthogonal position in the Chomsky hierarchy.

We have also present some works in this line. These works show that new challenging results in the field of GI can be obtained. Moreover, models in GI can be improved by using these linguistics ideas.

It is worth noting that a formal model bio-inspired by all these linguistic ideas could also help us to better understand natural language acquisition. By studying formal models of language learning, several key questions in linguistics can be answered, as for example, the type of input available to the learner, the impact of semantic information on learning the syntax of a language, etc. In fact, the model that we have proposed in [2] tries to answer some of these questions.

References

1. Angluin, D.: Learning regular sets from queries and counterexamples. Information and Computation 75, 87–106 (1987)
2. Angluin, D., Becerra-Bonache, L.: A model of semantics and corrections in language learning. In: YALEU/DCS/TR-1425 (April 2010)
3. Angluin, D., Becerra Bonache, L.: Experiments with an algorithm to learn meaning before syntax. In: ForLing2008, pp. 1–12 (2008)
4. Angluin, D., Becerra-Bonache, L.: Learning meaning before syntax. In: Clark, A., Coste, F., Miclet, L. (eds.) ICGI 2008. LNCS (LNAI), vol. 5278, pp. 1–14. Springer, Heidelberg (2008)
5. Becerra-Bonache, L.: On the Learnability of Mildly Context-Sensitive Languages using Positive Data and Correction Queries. PhD thesis, Rovira i Virgili University (2006)
6. Becerra-Bonache, L., Case, J., Jain, S., Stephan, F.: Iterative learning of simple external contextual languages. In: Freund, Y., Györfi, L., Turán, G., Zeugmann, T. (eds.) ALT 2008. LNCS (LNAI), vol. 5254, pp. 359–373. Springer, Heidelberg (2008)
7. Becerra-Bonache, L., de la Higuera, C., Janodet, J.C., Tantini, F.: Learning balls of strings from edit corrections. Journal of Machine Learning Research 9, 1841–1870 (2008)
8. Becerra-Bonache, L., Dediu, A.-H., Tîrnăucă, C.: Learning DFA from correction and equivalence queries. In: Sakakibara, Y., Kobayashi, S., Sato, K., Nishino, T., Tomita, E. (eds.) ICGI 2006. LNCS (LNAI), vol. 4201, pp. 281–292. Springer, Heidelberg (2006)
9. Becerra-Bonache, L., Yokomori, T.: Learning mild context-sensitiveness: Toward understanding children's language learning. In: Paliouras, G., Sakakibara, Y. (eds.) ICGI 2004. LNCS (LNAI), vol. 3264, pp. 53–64. Springer, Heidelberg (2004)
10. Bresnan, J., Kaplan, R.M., Peters, S., Zaenen, A.: Cross-serial dependencies in dutch. In: Savitch, W.J., Bach, E., Marsh, W., Safran-Naveh, G. (eds.) The Formal Complexity of Natural Language, pp. 286–319. D. Reidel, Dordrecht (1987)
11. Brown, R., Hanlon, C.: Derivational complexity and order of acquisition in child speech. In: Hayes, J.R. (ed.) Cognition and the Development of Language, pp. 11–54. Wiley, New York (1970)
12. Chouinard, M.M., Clark, E.V.: Adult reformulations of child errors as negative evidence. Journal of Child Language 30, 637–669 (2003)
13. Clark, A.: Grammatical inference and first language acquisition. In: Psychocomputational Models of Human Language Acquisition, Geneva, pp. 25–32 (2004)
14. Clark, E.V.: First Language Acquistion. Cambridge University Press, Cambridge (2002)
15. Culy, C.: The complexity of the vocabulary of bambara. In: Savitch, W.J., Bach, E., Marsh, W., Safran-Naveh, G. (eds.) The Formal Complexity of Natural Language, pp. 349–357 (1987)
16. de la Higuera, C.: A bibliographical study of grammatical inference. Pattern Recognition 38, 1332–1348 (2005)

17. de la Higuera, C.: Grammatical inference: learning automata and grammars. Cambridge University Press, Cambridge (2010)
18. Gold, E.M.: Language identification in the limit. Information and Control 10, 447–474 (1967)
19. Hirsh-Pasek, K., Treiman, R.A., Schneiderman, M.: Brown and hanlon revisited: mothers sensitivity to ungrammatical forms. Journal of Child Language 11, 81–88 (1984)
20. Joshi, A.K.: How much context-sensitivity is required to provide reasonable structural descriptions: Tree adjoining grammars. In: Dowty, D., Karttunen, L., Zwicky, A. (eds.) Natural Language Parsing: Psychological, Computational and Theoretical Perspectives, pp. 206–250. Cambridge University Press, Cambridge (1985)
21. Kudlek, M., Martín-Vide, C., Mateescu, A., Mitrana, V.: Contexts and the concept of mild context-sensitivity. Linguistics and Philosophy 26(6), 703–725 (2002)
22. Manaster-Ramer, A.: Some uses and abuses of mathematics in linguistics. In: Martín-Vide, C. (ed.) Issues in Mathematical Linguistics, pp. 73–130. John Benjamins, Amsterdam (1999)
23. Marcus, G.F.: Negative evidence in language acquisition. Cognition 46, 53–95 (1993)
24. Morgan, J.L., Travis, L.L.: Limits on negative information in language input. Journal of Child Language 16, 531–552 (1989)
25. Oates, T., Armstrong, T., Bonache, L.B., Atamas, M.: Inferring grammars for mildly context sensitive languages in polynomial-time. In: Sakakibara, Y., Kobayashi, S., Sato, K., Nishino, T., Tomita, E. (eds.) ICGI 2006. LNCS (LNAI), vol. 4201, pp. 137–147. Springer, Heidelberg (2006)
26. Parekh, R., Honavar, V.: Grammar inference, automata induction and language acquisition. In: Moisl Dale and Somers, editors, pp. 727–774. Marcel Dekker, New York (2000)
27. Pinker, S.: Formal models of language learning. Cognition 7, 217–283 (1979)
28. Sakakibara, Y.: Recent advances of grammatical inference. Theoretical Computer Science 185, 15–45 (1997)
29. Schaerlaekens, A.M.: The two-word sentence in child language development. In: Mouton, The Hague (1973)
30. Schlesinger, I.M.: Production of utterances and language acquisition. In: Slobin, D.I. (ed.) The Ontogenesis of Grammar, pp. 63–103. Academic Press, New York-London (1971)
31. Shieber, S.M.: Evidence against the context-freeness of natural languages. In: Savitch, W.J., Bach, E., Marsh, W., Safran-Naveh, G. (eds.) The Formal Complexity of Natural Language, pp. 320–334. D. Reidel, Dordrecht (1987)
32. Valiant, L.G.: A theory of the learnable. Communication of the ACM 27, 1134–1142 (1984)
33. Wexler, K., Culicover, P.: Formal Principles of Languages Acquisition. MIT Press, Cambridge (1980)

Differential Evolution for Protein Structure Prediction Using the HP Model

J. Santos and M. Diéguez

Computer Science Department, University of A Coruña, Spain
{jose.santos,martin.dieguez}@udc.es

Abstract. We used Differential Evolution (DE) for the problem of protein structure prediction. We employed the HP model to represent the folding conformations of a protein in a lattice. In this model the nature of amino acids is reduced considering only two types: hydrophobic residues (H) and polar residues (P), which is based on the recognition that hydrophobic interactions are a dominant force in protein folding. Given a primary sequence of amino acids, the problem is to search for the folding structure in the lattice that minimizes an energy potential. This energy reflects the fact that the hydrophobic amino acids have a propensity to form a hydrophobic core. The complexity of the problem has been shown to be NP-hard, with minimal progress achieved in this category of ab initio folding. We combined DE with methods to transform illegal protein conformations to feasible ones, showing the capabilities of the hybridized DE with respect to previous works.

1 Brief Introduction to the Problem of Protein Structure Prediction

Proteins only attain their sophisticated catalytic activities by folding into complex 3D structures. The amino acid hydrophobicity plays an important role in defining this folding process: for reasons of thermodynamics, most proteins fold with hydrophobic side-chains pointing inwards to form a hydrophobic interior and present a hydrophilic surface to the watery cytoplasm of the cell. Anfinsen [1] showed that a protein in its natural environment folds into a unique three dimensional structure, the native structure, independent of the starting conformation. The thermodynamic hypothesis states that the native conformation of the protein is the one with lowest Gibbs free energy. That native state of a protein plays an essential role in the functionality of a protein. As experimental determination of the native conformation is still difficult and time consuming, much work has been done to forecast the native conformation computationally.

There is an extensive research done on the direct prediction of the final protein structure conformations (secondary and tertiary structures). In the case of secondary structure prediction (local regular elements such as helixes and strands), machine learning methods such as neural nets and support vector machines, can achieve up 80% overall accuracy in globular proteins [7].

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 323–333, 2011.

In the case of the prediction of the final tertiary structure, the methods range from comparison methods with resolved structures to the "ab initio" prediction. In the first case, the search space is pruned by the assumption that the target protein adopts a structure close to the experimentally determined structure of another homologous protein. But the output of experimentally determined protein structures -by time-consuming and relatively expensive X-ray crystallography or NMR spectroscopy- is lagging far behind the output of protein sequences. Because of this, the most difficult *ab initio* prediction is a challenge in bioinformatics. It uses only the information from the amino acid sequence of the primary structure [21]. In such prediction there are models that simplify the complexity of the interactions and the nature of the amino acid elements, like the models that locate these in a lattice, or detailed atomic models like the Rosetta system [20]. Nevertheless, as Zhao [24] indicates, such detailed atomic models would not be able to explore more than small changes that occur over very small timescales and they involve many parameters and approximations. For this reason, simplified or minimalist models are employed. The use of a reduced alphabet of amino acids is based on the recognition that the binary pattern of hydrophobic and polar residues is a major determinant of the folding of a protein.

In the HP model [6] the elements of the chain can be of two types: H (hydrophobic residues) and P (polar residues). The sequence is assumed to be embedded in a lattice that discretizes the space conformation and can exhibit different topologies such as 2D square or triangular lattices, or 3D cubic or diamond lattices. The interaction between two H elements that are adjacent in the lattice (and not consecutive in the primary sequence) is -1 and zero for the other possible pairs. That is, the HP energy matrix only implies attractions (H with H), and neutral interactions (P with P and P with H). Given a primary sequence, the problem is to search for the folding structure in the lattice that minimizes the energy. The complexity of the problem has been shown to be NP-hard [10,23] and the progress was slow; as Unger points out "minimal progress was achieved in the category of ab initio folding" [22]. Although the HP model is simple, it is powerful enough to capture many properties of actual proteins. It is non-trivial, captures many global aspects of real proteins and still remains the hardness features of the original biological problem [8]. For this reason, many authors have been working on several evolutionary algorithms [22,24] in the direct prediction of the native conformations using the HP model, as we detail in the next section.

Additionally, we must take into account that the energy landscape in this problem presents a multitude of local energy minima separated by high barriers. As Zhao indicates "there are many meta-stable states whose energies are very close to the global minimum and an exceedingly small number of global optimal states. Folding energy landscapes are funnel-like" [24]. For this reason, we will test the capability of Differential Evolution as a method with a better control in the balance between exploration and exploitation with respect to a classical genetic algorithm, as detailed in Section 3. Moreover, we will introduce methods to translate illegal protein conformations to feasible ones, smoothing the landscape (Sect. 3.2). Finally, we will test our proposals with benchmark series (Sect. 4).

2 Previous Work with Evolutionary Algorithms

As we commented, many authors have been working on several evolutionary algorithms, especially genetic algorithms (GA), to determine the final folded conformations using the HP model [22]. One of the main decisions is the encoding of the problem which has to be optimized, in this case, the genotypic encoding of the protein conformation in the lattice. Three basic possibilities can be considered to the representation of the folded sequence in the lattice: Cartesian coordinates and two alternatives with internal coordinates.

In the first case, the location of each amino acid is specified independently with its Cartesian coordinates. With the internal coordinates the embedding of the protein is specified as a sequence of moves taken on the lattice from one amino acid to the next. The first alternative with internal coordinates uses an absolute representation [23] and moves are specified with respect to it. For example, in the case of the square lattice: North, South, East and West. A conformation is expressed as a sequence $\{N, S, E, W\}^{n-1}$, which is the genetic material in the individuals when this representation is used. In the relative representation, relative moves are considered. The reference system is not fixed and the next move depends on the previous move. Now, in the same case as before, three moves are allowed: Forward, Turn Left and Turn Right. The conformations are expressed now as sequences $\{F, L, R\}^{n-1}$. This representation has the advantage of guaranteeing that all solutions are 1-step self-avoiding (because there is no back move).

2.1 Initial Population, Space of Feasible Conformations and Fitness Alternatives

Unger and Moult [23] only considered feasible conformations in the genetic population, that is, embeddings that are self-avoiding paths on the considered lattice. In their work, when operators such as mutation and crossover were applied, the GA iterated until a “nonlethal” (in authors’ words) conformation was generated. Similarly, Rylance [17] obtained the initial population using a recoil growth algorithm. Song et al. [19] applied the genetic operators used in [3], with 6 types of mutations; according to the authors, one of them (cornerchange, which mutates one shape to another shape) makes the conformation has more biological significance.

Patton et al. [15] used the relative representation. In addition, unlike the previous authors, they allowed illegal conformations, but using a penalty in the fitness to avoid self-avoiding constraints. This essentially allows the search to proceed through illegal states. With these modifications the results were a bit better in terms of minimization of energy and required computational steps. Krasnogor et al. [12] analyzed the impact of different factors when evolutionary algorithms are used to the problem: the conformational representation, the energy formulation and the way in which infeasible conformations are penalized. Their results supported the use of the relative encoding.

Cotta [2] explored alternatives to the penalty approach. He used a genetic algorithm hybridized with a backtracking algorithm or repair procedure that maps infeasible solutions to feasible conformations. The combination provided slight better results, with respect to a penalty-based approach and to a feasible-space approach. Custódio et al. [4] included a modification in the usual fitness function. The modification was based on the assumption that it may be preferable for a hydrophobic monomer to have a polar neighbor than to be in direct contact with the polar solvent. Their results suggest that the modified model has a greater tendency to form globular structures. Lopes and Scapin [14] used a term in the fitness function, radius of gyration, which indicates how compact a set of amino acids is. Hence, more compact conformations are rewarded.

2.2 Other Alternatives of Optimization

Within the natural or bioinspired computing field different algorithms have been used. For instance, Shmygelska and Hoos [18] presented the use of an ant colony optimization (ACO) algorithm for the 2D HP model and its extension to the 3D HP model, with the inclusion of a local search, that is, the virtual ants further optimized protein conformations. With a pheromone updating procedure, ants folded a protein adding one amino acid at a time based on the pheromone matrix value, which represents previous experience of previous ants, together with the use of heuristic information. The local greedy search introduced long range mutations in the chain. The algorithm scaled worse with sequence length.

Krasnogor et al. [11] also used multimeme algorithms for the problem. The term multimeme means that a set of local searches are used in the individuals of the genetic population. The authors remarked the robustness of the approach across a range of protein structure models and instances. More recently, Lopes and Bitello [13] applied differential evolution for the 2D HP problem. The authors reported that the DE approach was much more consistent than the GA algorithm, since it achieved the maximum number of HH contacts in almost all runs when several benchmark sequences were used. Finally, Cutello et al. [5] proposed an immune algorithm with two special mutation operators, hypermutation and hypermacromutation, which were incorporated into the proposed algorithm to allow effective searching.

3 Methods. Differential Evolution

Differential Evolution (DE) [16] is a population-based search method. DE creates new candidate solutions by combining existing ones according to a simple formula of vector crossover and mutation, and then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand. The central idea of the algorithm is the use of difference vectors for generating perturbations in a population of vectors. This algorithm is specially suited for optimization problems where possible solutions are defined by a real-valued vector. The basic DE algorithm is summarized in the pseudo-code of Algorithm 1.

Algorithm 1. Differential Evolution Algorithm

```
1: Initialize the population randomly
2: repeat
3:   for all individual $x$ in the population do
4:     Let $x_1, x_2, x_3 \in$ population, randomly obtained $\{x_1, x_2, x_3, x$ different from each other.$\}$
5:     Let $R \in \{1, ..., n\}$, randomly obtained {n is the length of the chain.}
6:     for $i = 1$ to $n$ do
7:       Pick $r_i \in U(0, 1)$ uniformly from the open range (0,1).
8:       if $(i = R) \vee (r_i < CR)$ then
9:         $y_i \leftarrow x_{1i} + F(x_{2i} - x_{3i})$
10:      else
11:        $y_i = x_i$
12:      end if
13:    end for{$y = [y_1, y_2 ... y_n]$ is a new generated candidate individual}
14:    if $f(y) < f(x)$ then
15:      Replace individual $x$ by $y$
16:    end if
17:  end for
18: until termination criterion is met
19: return $z \in$ population $\backslash \forall t \in$ population , $f(z) <= f(t)$
```

One of the reasons why Differential Evolution is an interesting method in many optimization or search problems is the reduced number of parameters that are needed to define its implementation. The parameters are F or differential weight and CR or crossover probability. The weight factor F (usually in $[0, 2]$) is applied over the vector resulting from the difference between pairs of vectors (x_2 and x_3). CR is the probability of crossing over a given vector of the population (x) and a candidate vector (y) created from the weighted difference of two vectors ($x_1 + F(x_2 - x_3)$). Finally, the index R guarantees that at least one of the parameters (genes) will be changed in such generation of the candidate solution.

The main problem of the genetic algorithm (GA) methodology is the need of tuning of a series of parameters: probabilities of different genetic operators such as crossover or mutation, decision of the selection operator (tournament, roulette,...), tournament size. Hence, in a standard GA it is difficult to control the balance between exploration and exploitation. On the contrary, DE reduces the parameters tuning and provides an automatic balance in the search. As Feoktistov [9] indicates, the fundamental idea of the algorithm is to adapt the step length ($F(x_2 - x_3)$) intrinsically along the evolutionary process. At the beginning of generations the step length is large, because individuals are far away from each other. As the evolution goes on, the population converges and the step length becomes smaller and smaller.

In addition to the usual implementation of DE we used these two strategies: i) Regarding the weighted differences (line 9 in pseudo-code), we used 4 different vectors to calculate the candidate solutions, because, as indicated by Price et al. [16], for large populations four vectors are more effective regarding convergence than the weighted difference of only two vectors. ii) Moreover, the usual implementation of DE chooses the base vector x_1 randomly or as the individual with the best fitness found up to the moment (x_{best}). To avoid the high selective pressure of the latter, the usual strategy is to interchange the two possibilities across generations. Instead of this, we used a tournament to pick the vector x_1, which allows us to easily establish the selective pressure by means of the tournament size.

3.1 Encoding and Fitness

Individuals in DE are real-valued vectors which, in turn, are decoded into a specific fold of an amino acid chain in the square lattice. We used the same real-valued encoding proposed by Lopes and Bitello [13] with relative coordinates. In the bi-dimensional lattices there are only three possible moves: (F)orward, (R)ight and (L)eft. Therefore, the phenotypical representation of a solution is defined over the alphabet F, R, L. The genotype that represents such phenotypical protein conformation is a real-valued vector X. Considering X_{ij} the $j-th$ element of vector X_i, and P the string representing the sequence of moves of the folding, and $\alpha < \beta < \delta < \gamma$ arbitrary constants in R, the genotype-phenotype mapping is defined as follows:

$$\begin{array}{l} If\ \alpha < X_{ij} \leq \beta \ \ then\ P_j = L \\ If\ \beta < X_{ij} < \delta \ \ then\ P_j = F \\ If\ \delta \leq X_{ij} < \gamma \ \ then\ P_j = R \end{array} \tag{1}$$

We used $\alpha = -3$, $\beta = -1$, $\delta = 1$ and $\gamma = 3$, as in [13]. If a component of a mutant vector (candidate solution) goes off its limits, as consequence of the DE formula for each candidate solution (line 9 in the pseudo-code), we set the component to a random value in the corresponding interval of the bound limit. Note that the mapping allows several genotypes to represent a single phenotype.

We used initial "stretched" individuals (all elements of the string are F) when creating the initial population. The procedure ensures initial individuals with diversity in the genotypic level, as for the Fs are used random values in the corresponding interval.

Finally, we used the simple fitness function that considers only the maximization of H-H contacts. The infeasible solutions are given a fitness of 0.

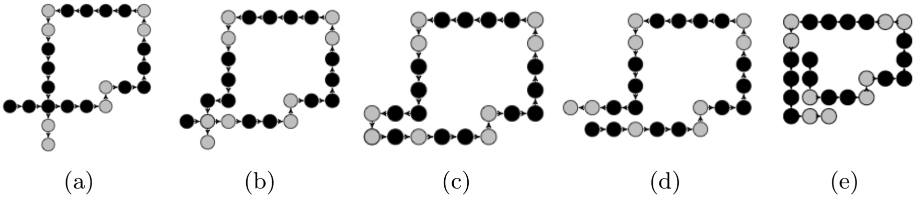


Fig. 1. Repair processes. (a) Original illegal conformation; (b-c) and (d) Intermediate steps and final conformation with the absolute moves procedure; (e) Final conformation with the procedure which uses the Cartesian coordinates.

3.2 Repair Process

This process takes as input an infeasible protein folding (more than one amino acid in the same lattice location) and tries to repair it to a feasible one, trying to maintain most of the original illegal conformation. We implemented two different strategies. The first one works in the Cartesian coordinate space: From an illegal folding, it processes the chain across its amino acids and, if a conflict is encountered, it moves the amino acid that generates the conflict to the nearest free position. In a second stage, the subsequent chain is reconstructed from the new obtained positions, respecting the restrictions of the lattice model.

The second strategy works with the absolute moves (North, South, East and West in the square lattice) to repair the chain conformation. Once a conflict is found, it is changed the absolute move of the amino acid that generates the conflict to obtain a position with no conflict. The procedure is repeated through the rest of the protein conformation. Nevertheless, we did not use a complete search as we did not perform backtracking strategies when there is not any free position after checking all the possible moves. Figure 1 shows an example, where Fig. 1.a is the illegal conformation, Fig. 1.b and Fig. 1.c are intermediate states of this repair procedure and Fig. 1.d is the final feasible conformation. Figure 1.e represents the final rectified conformation from the illegal one of Fig. 1.a using the first procedure with the Cartesian coordinates.

The first strategy tries to obtain legal conformations searching for a similar one in the Cartesian space, while the second tries to maintain the relative conformation of the rest of the chain which has not generated the conflict. In both cases, when there is no possibility of a repair process, the illegal conformation remains in the population (with fitness 0, so it will soon disappear).

Table 1. Benchmarks sequences, for the square lattice, used in the experiments

	Length	**HP Chain**	**E**
S1	20	$HPHP^2H^2PHP^2HPH^2P^2HPH$	-9
S2	24	$H^2P^2HP^2HP^2HP^2HP^2HP^2HP^2H^2$	-9
S3	25	$P^2HP^2H^2P^4H^2P^4H^2P^4H^2$	-8
S4	36	$P^3H^2P^2H^2P^5H^7P^2H^2P^4H^2P^2HP^2$	-14
S5	48	$P^2HP^2H^2P^2H^2P^5H^{10}P^6H^2P^2H^2P^2HP^2H^5$	-23
S6	50	$H^2PHPHPHPH^4PHP^3HP^3HP^4HP^3HP^3HPH^4PHPHPHPH^2$	-21
S7	60	$P^2H^3PH^8P^3H^{10}PHP^3H^{12}P^4H^6PH^2PHP$	-36
S8	64	$H^{12}PHPHP^2H^2P^2H^2P^2HP^2H^2P^2H^2P^2HP^2H^2P^2H^2P^2HPHPH^{12}$	-42

4 Results

We tested the DE implementation hybridized with the repair processes using benchmark sequences employed in the literature. Table 1 shows these sequences with different number of amino acids and the corresponding energy minima (maximum number of HH contacts known up to now).

Regarding the DE implementation, we used standard values: $CR = 0.9$ and $F = 0.9$, whereas the size of the tournament to choose the base vector was 8% of the population, which implies a low selective pressure when choosing such vector to disturb.

Regarding the repair processes, as there is not a clear rule to determine what the best strategy is, we used the two repair processes implemented. The first one, which works with the Cartesian coordinates, is used beginning the conflict checking in the first amino acid of the protein chain as well as starting it from the center of the chain. The second repair strategy, with absolute moves, always begins from the first amino acid. These three possibilities were used with the same probability when the candidate solution used by DE generates an illegal protein conformation.

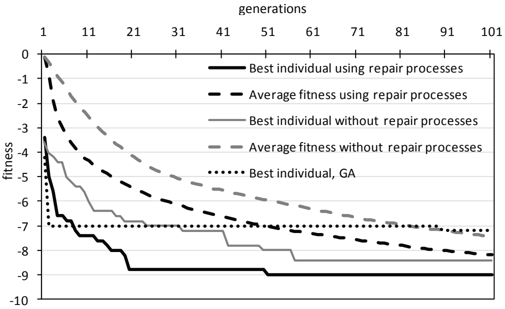


Fig. 2. Comparison of DE evolution with and without the use of the repair processes

Figure 2 shows a comparison between the two alternatives, using DE with and without the repair processes. The protein series used was one of the benchmark sequences shown in Table 1 (sequence *S2*). The quality evolutions in the Figure are an average of 5 different evolutions with different initial populations. As the repair processes allow that almost all the individuals in the population are feasible, DE combined with them obtains the optimum configuration in fewer generations. DE without the repair processes needs much more than the 100 generations shown in the Figure to obtain the best configuration in all the runs. For the comparison purpose, we also included in the Figure the evolution of the average quality of the best individual of 5 different runs of a GA with typical parameters (one-point crossover with probability of 0.5, mutations with a probability of 0.02, tournament selection with a size of 4% of the population, and use of the relative encoding).

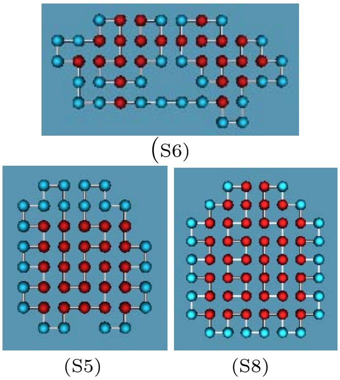


Fig. 3. Optimal configurations for three of the sequences: S5, S6 and S8

In Table 2 we included the comparison of our hybrid DE solution with the results of Unger and Moult with a GA [23], using the benchmark series for the square lattice. In the table, E_{min} represents the maximum number of HH contacts. Previous works summarize the results taking into account independent runs for each protein sequence, and in each column it is specified the best energy value found in the different runs of the corresponding search algorithm. If the best known solution quality was found, it is indicated in bold. The values in parentheses are the minimum number of conformations scanned before the lowest energy values were found in one of the runs. This is the case of the results of Unger and Moult [23] from 5 independent runs. In our case, we included the average energy in the different runs and the average number of

Table 2. Comparison of results with the benchmark sequences

Seq.	E_{min}	U&M GA [23]	hybrid DE	L&B DE [13]	ACO [18]
S1	-9	**-9** (30492)	**-9**,-9 (3584, 6362)	**-9**,-9	**-9**
S2	-9	**-9** (30491)	**-9**,-9 (5806, 9292)	**-9**,-9	**-9**
S3	-8	**-8** (20400)	**-8**,-8 (7061, 18828)	**-8**,-8	**-8**
S4	-14	**-14** (301339)	**-14**,-14 (45793, 92579)	**-14**,-13.96	**-14**
S5	-23	-22 (126547)	**-23**,-23 (245943, 532787)	**-23**,-23	**-23**
S6	-21	**-21** (592887)	**-21**,-21(365222, 691989)	**-21**,-21	**-21**
S7	-36	-34 (208781)	-35,-33.57	-35,-34.79	**-36**
S8	-42	-37 (187393)	**-42**,-42 (176313, 340917)	**-42**,-41.87	**-42**

evaluations that obtained the best optimum value (second values after ","). We used 10 independent runs for each sequence.

There is not a clear rule about what the best number of individuals is, so, for each sequence, we used *population size = number of amino acids x 15* (as suggested in [16] and [13]). One of the main factors to be compared is the number of necessary evaluations to find the optimum. The work of Lopes and Bitello did not include the number of evaluations to find their best values [13], so we only included in the Table the best and average energy values reported by the authors. The hybrid DE with the inclusion of the repair procedures reduces significantly the number of conformations that are needed to scan in order to obtain the best energy values, as it can be seen with such best minimum number of necessary evaluations (first values in parentheses). Even the average values of the number of necessary scanned conformations is lower with respect to the reported values of Unger and Moult with a GA [23] (except for S6), taking into account that their reported values are the best in 5 independent runs. Figure 4 shows three examples of the best folded conformations obtained with the hybrid DE. For the sequence $S7$ the minimum was not obtained, although we allowed a maximum number of 10^6 evaluations in the different runs.

In the case of Shmygelska and Hoos [18], using an ant colony optimization (ACO), the authors obtained the best values in all the sequences. Their results are difficult to compare with the evolutionary algorithms, as the authors used 500 runs of the ACO algorithm (in sequences $S1 - S6$) and 100 runs for sequences $S7$ and $S8$, with a population of 100 ants and, additionally, half of the ants performed a local search with a maximum of 1,000 scans through the protein sequence ($S1 - S6$) or a maximum of 10,000 scans ($S7$ and $S8$). As the authors commented, their empirical results indicated that their "rather simple ACO algorithm scales worse with sequence length but usually finds that more diverse ensemble of native states".

5 Conclusions

We applied DE to the optimization of folded protein conformations using the HP lattice model. DE provides a method with a reduced number of defining parameters in its implementation. The differences of vectors used in DE are adequate to search in the multimodal energy landscape inherent to the HP protein folding problem, where the adaptive perturbations or mutations are applied over the segments of the protein conformations that are different in the vectors or individuals of the genetic population. The hybridization of DE with the repair procedures provided a significant reduction of the necessary conformations to obtain the best energy values in different benchmark sequences.

Acknowledgments. This paper has been funded by the Ministry of Science and Innovation of Spain (project TIN2007-64330).

References

1. Anfinsen, C.B.: Principles that govern the folding of proteins. Science 181(96), 223–230 (1973)
2. Cotta, C.: Protein structure prediction using evolutionary algorithms hybridized with backtracking. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2687, pp. 321–328. Springer, Heidelberg (2003)
3. Cox, G.A., Mortimer-Jones, T.V., Taylor, R.P., Johnston, R.L.: Development and optimization of a novel genetic algorithm for studying model protein folding. Theor. Chem. Acc. 112, 163–178 (2004)
4. Custódio, F.L., Barbosa, H.J.C., Dardenne, L.E.: Investigation of the three dimensional lattice HP protein folding model using a genetic algorithm. Genetics and Molecular Biology 27(4), 611–615 (2004)
5. Cutello, V., Nicosia, G., Pavone, M., Timmis, J.: Immune algorithm for protein structure prediction on lattice models. IEEE T. Evol. Comp. 11(1), 101–117 (2007)
6. Dill, K.A.: Dominant forces in protein folding. Biochemestry 29, 7133–7155 (1990)
7. Dor, O., Zhou, Y.: Achieving 80% tenfold cross-validated accuracy for secondary structure prediction by large-scale training. Proteins 66(4), 838–845 (2007)
8. Dill, K.A.: et al. Principles of protein folding: a perspective from simple exact models. Protein. Science 4(3), 561–602 (1995)
9. Feoktistov, V.: Differential Evolution: In Search of Solutions. Springer, Heidelberg (2006)
10. Hart, W.E., Istrail, S.: Robust proofs of NP-hardness for protein folding: General lattices and energy potentials. Journal of Computational Biology 4(1), 1–22 (1997)
11. Krasnogor, N., Blackburne, B.P., Burke, E.K., Hirst, J.D.: Multimeme algorithms for protein structure prediction. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 769–778. Springer, Heidelberg (2002)
12. Krasnogor, N., Hart, W.E., Smith, J., Pelta, D.A.: Protein structure prediction with evolutionary algorithms. In: Proc. GECCO 1999, pp. 1596–1601 (1999)
13. Lopes, H.S., Bitello, R.A.: Differential evolution approach for protein folding using a lattice model. J. of Computer Science and Technology 22(6), 904–908 (2007)
14. Lopes, H.S., Scapin, M.P.: An enhanced genetic algorithm for protein structure prediction using the 2D hydrophobic-polar model. In: Talbi, E.-G., Liardet, P., Collet, P., Lutton, E., Schoenauer, M. (eds.) EA 2005. LNCS, vol. 3871, pp. 238–246. Springer, Heidelberg (2006)

15. Patton, W.P., Punch, W.F., Goldman, E.: A standard genetic algorithm approach to native protein conformation prediction. In: Proceedings of 6th International Conference on Genetic Algorithms, pp. 574–581 (1995)
16. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution. A Practical Approach to Global Optimization. Springer - Natural Computing Series (2005)
17. Rylance, G.J.: Applications of genetic algorithms in protein folding studies, PhD Thesis. University of Birmingham (2004)
18. Shmygelska, A., Hoos, H.H.: An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. Bioinformatics 6, 30 (2005)
19. Song, J., Cheng, J., Zheng, T., Mao, J.: A novel genetic algorithm for HP model protein folding. In: Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies, pp. 935–937 (2005)
20. Rosetta system, http://www.rosettacommons.org
21. Tramontano, A.: Protein structure prediction. Concepts and applications. Wiley-VCH (2006)
22. Unger, R.: The genetic algorithm approach to protein structure prediction. Structure and Bonding 110, 153–175 (2004)
23. Unger, R., Moult, J.: Genetic algorithms for protein folding simulations. Journal of Molecular Biology 231(1), 75–81 (1993)
24. Zhao, X.: Advances on protein folding simulations based on the lattice HP models with natural computing. Applied Soft Computing 8, 1029–1040 (2008)

A Preliminary Study on the Prediction of Human Protein Functions

Guido Bologna[1], Anne-Lise Veuthey[2], Marco Pagni[3],
Lydie Lane[1,4], and Amos Bairoch[1,4]

[1] CALIPHO Group, Swiss Institute of Bioinformartics
Rue Michel Servet 1, 1211 Geneva 4, Switzerland
Guido.Bologna@isb-sib.ch, Lydie.Lane@isb-sib.ch
[2] Swiss-Prot Group, Swiss Institute of Bioinformartics
Rue Michel Servet 1, 1211 Geneva 4, Switzerland
Anne-Lise.Veuthey@isb-sib.ch
[3] Vital-IT Group, Swiss Institute of Bioinformartics
Quartier Sorge - Genopode, 1015, Switzerland
Marco.Pagni@isb-sib.ch
[4] Department of Structural Biology and Bioinformatics, University of Geneva
Rue Michel Servet 1, 1211 Geneva 4, Switzerland
Amos.Bairoch@unige.ch

Abstract. In the human proteome, about 5'000 proteins lack experimentally validated functional information. In this work we propose to tackle the problem of human protein function prediction by three distinct supervised learning schemes: one-versus-all classification; tournament learning; multi-label learning. Target values of supervised learning models are represented by the nodes of a subset of the Gene Ontology, which is widely used as a benchmark for functional prediction. With an independent dataset including very difficult cases the recall measure reached a reasonable performance for the first 50 ranked predictions, on average; however, average precision was quite low.

1 Introduction

The recent completion of the annotation of the human proteome revealed that out of the 20'400 protein-coding genes that exist in the human genome, about 5'000 of them lack experimentally validated functional information. To date, many computational methods have been developed to predict gene function [10]. These methods do not exclusively rely on sequence homology and domain composition but also integrate contextual information, which is beyond the protein sequence itself. Within this category, algorithms for protein function prediction have been developed based on genomic context [1], protein-protein interaction networks [17], and phenotypic profiling [11].

In this work we propose to tackle the problem of human protein function prediction by three distinct supervised learning schemes : one-versus-all classification; tournament learning; multi-label learning (MLL). The target classes

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 334–343, 2011.

provided to the predictors correspond to terms of a *Gene Ontology* (GO) [12]. GO provides a structured description of the possible functions of genes or proteins and is classically used as a benchmark for functional prediction.

In [7] neural networks and SVMs were proposed to predict human protein functions related to 14 GO terms. Based on five-fold cross-validation trials the average recall was 50% and the average specificity above 90%. In [16] multi-label learning was applied to the classification of yeast gene function prediction with target classes defined by GO terms. With the use of neural networks the average precision during cross-validation trials was 75.6%. Vens *et al.* adapted decision trees to the multi-label learning scheme on yeast gene function prediction [15]. Target classes were defined by the *FunCat* hierarchy classification [9] and GO terms. They concluded that learning one single model for all classes gave better average accuracy than learning classes, independently. Based on the nearest neighbour classifier and 24 functional classes, the authors of [6] obtained average predictive accuracy for mouse gene function prediction equal to 69.1%. In [10] a standardised collection of mouse functional genomic datasets were assembled. Nine bioinformatics teams independently trained classifiers and generated predictions of GO terms. The conclusion was that at a recall rate of 20%, a unified set of predictions averaged 41% precision. On yeast, based on support vector machines predictors and a Bayesian network the average recall and the average precision obtained on an independent dataset were 7%, and 51%, respectively on GO terms [2]. In [4] the SVMProt predictor was presented. The authors focused on a number of protein classes collected from several databases that encompassed all major classes of enzymes, receptors, transporters, channels, DNA-binding proteins and RNA-binding proteins. The obtained accuracy for protein family classification was found to be in the range of 69.1%–99.6%.

To the best of our knowledge no other work has explored the prediction of human protein functions with the use of GO targets learned by tournament learning (also denoted as *one-versus-one* learning) and multi-label learning [13]. On cross validation trials performed with the use of multi-label learning applied to neural networks the average precision was 63.4%. With an independent dataset including very difficult cases the recall measure reached a reasonable performance for the first 50 ranked predictions, on average; however, average precision was quite low. In the following sections we introduce the methods applied to predict protein functions then we describe three series of experiments, followed by a conclusion.

2 Models and Methods

In this work we compare several learning models : decision trees [3], SVMs [14], and neural networks trained by multi-label learning [16]. As SVMs are binary classifiers, a classification problem of K classes (with $K > 2$) is transformed into K binary classification problems learned by K classifiers. For each classifier there is a positive class and a negative class encompassing the other $K - 1$ classes. This classification scheme is denoted as *one-versus-all* classification. In tournament learning all the possible couples of two classes are learned. Thus, with K independent classes there are $((K-1) \cdot K)/2$ binary classification problems.

Traditional learning algorithms used by neural networks, SVMs or decision trees learn to discriminate instances by taking into account that classes will be independent. Given that a protein can have more than a particular annotated function, the traditional learning scheme of the proposed Machine Learning models is not completely appropriate.

In Multi-Label Learning with neural networks, the traditional sum of squares' error function has been modified to take into account the correlations between several target classes [16]. As well as neural networks, decision trees and SVMs were recently adapted to take into account multi-label learning [8], [15].

3 The Human Proteome Dataset

The number of human proteins extracted from the UniProtKB/Swiss-Prot database (UniProtKB release 15.13) was 20'228. An independent testing set of 44 "newly functionalised" proteins has been extracted from a more recent version of this database (UniProtKB release 2011_1).

3.1 Input Vectors

The input vectors of the human proteome dataset were generated according to a number of features pertaining to the protein sequence. Except for protein interactions all the input features result from predictors. Table 1 presents the list of these features. Each protein is represented by 33'102 input variables. More than 20'000 Boolean input variables are due to protein/protein interactions extracted from String. InterPro domains represent more than 7'000 inputs, and PROSITE patterns/profiles give more than 5'000 inputs. As a result, input vectors are essentially sparse.

3.2 Output Vectors

The target vectors are defined according to the Gene Ontology (GO), which is a major bioinformatics initiative to unify the representation of gene and gene product attributes across all species. GO is represented as a directed acyclic graph covering three distinct sections:

1. Molecular function ("F")
2. Biological process ("P")
3. Cellular component ("C")

GO nodes at the top of the structure are the most general, whereas at the bottom lie very specific properties. For instance, for the "F" branch we have at the top "*molecular function*" and several levels below there is "*Hydrolase activity acting on ester bonds*", which represents a more specific function of "*Molecular Function*".

From this ontology we selected all the GO terms which annotate from 20 up to 3'000 human proteins. Note that we discarded all the proteins that were associated to the GO terms by electronic annotation, because we do not want to rely on GO annotations that have been predicted on the basis of features that are part of our input vectors (for example InterPro domains). Finally, we obtained 1'930 GO terms representing output targets for the supervised models to be trained. Since a protein can have more than a function, an input vector can be associated to more than a GO term.

Table 1. Input features

Protein Feature	Internet Location
InterPro domains	www.ebi.ac.uk/interpro/
PROSITE patterns/profiles	expasy.org/prosite/
Coiled coil domains	www.ch.embnet.org/software/COILS_form.html
Transmembrane domains	In-house program following [5]
Transmembrane domains	saier-144-21.ucsd/memsat.html
Transmembrane domains	www.cbs.dtu.dk/services/TMHMM/
GPI lipid anchors	mendel.imp.ac.at/gpi/gpi_server.html
N-Glycosylation sites	www.cbs.dtu.dk/services/NetNGlyc/
O-glycosylation sites	www.cbs.dtu.dk/services/NetOGlyc/
N-terminal myristoylation	mendel.imp.ac.at/myristate/SUPLpredictor.htm
N-terminal myristoylation	expasy.org/tools/myristoylator/
Presence of poly-aminoacids	In-house program
Sequence repeats	www.embl.de/∼andrade/papers/rep/search.html
Signal peptide cleavage sites	www.cbs.dtu.dk/services/SignalP/
Tyrosine sulfation sites	expasy.org/tools/sulfinator/
Subcellular location and cleavage site	www.cbs.dtu.dk/services/TargetP/
Disordered regions	anchor.enzim.hu/
Interaction with other human proteins	string-db.org/

4 Experiments

In the first series of experiments, we took into account 1930 distinct classification problems, for which positive examples represent proteins annotated by a particular GO term, while the negative examples are selected outside the branch of its associated GO term. This classification strategy corresponds to the "*one-versus-all*" approach. We performed both cross-validation trials and tests on the independent testing set.

In the second series of experiments we used a different training strategy denoted as “tournament learning”. On the 359 GO terms of the “F” sub-ontology we selected disjointed pairs of GO terms (i.e. pairs of GO terms having no proteins in common), then we trained the resulting 46’564 predictors and finally we tested them on the independent testing set. Finally, in the third series of experiments we applied the MLL learning scheme to the 359 GO terms of the “F” sub-ontology.

4.1 One-versus-All Experiments

Within this framework very often the number of negative instances is much greater than the number of positive instances. In other words, classes with a small proportion of proteins may be difficult to predict. For that reason we decided to perform many evaluation trials by varying the proportion of negative examples. The models we used during the experiments were linear SVMs provided by the Matlab Bioinformatics toolbox. Learning parameters were set to default values and results were evaluated on recall ($= tp/(tp + fn)$; tp = true positives; fn = false negatives), precision ($= tp/(tp + fp)$; fp = false positives) and specificity ($= tn/(tn+fp)$; tn = true negatives). Figure 1 illustrates average recall, average precision and average specificity by varying the proportion factor of negative proteins from 1 to 4. The points of the curves represent average values obtained after 10 repetitions of 10-fold cross-validation over all the 1930 GO terms.

On 1930 balanced classification problems we compared SVMs with CART decision trees. We used the Statistic Toolbox of Matlab by setting the learning parameters with default values. Figure 2 shows the histogram of average accuracy differences obtained by 10 repetitions of 10 fold cross-validation (for all the datasets). For SVMs the average of the average accuracies was 78.9%, while for

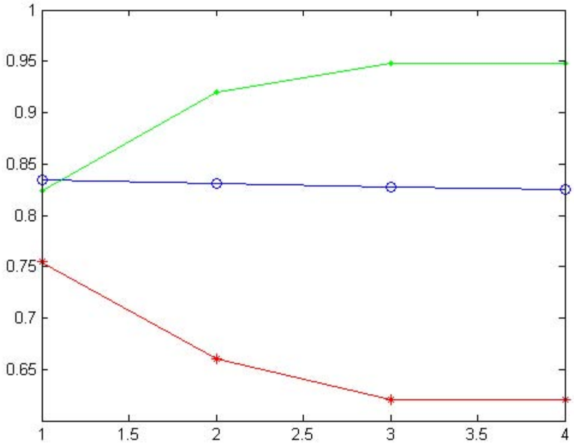


Fig. 1. From bottom to top : average recall, average precision and average specificity by varying the proportion factor of negative proteins from 1 to 4 over 1930 GO terms

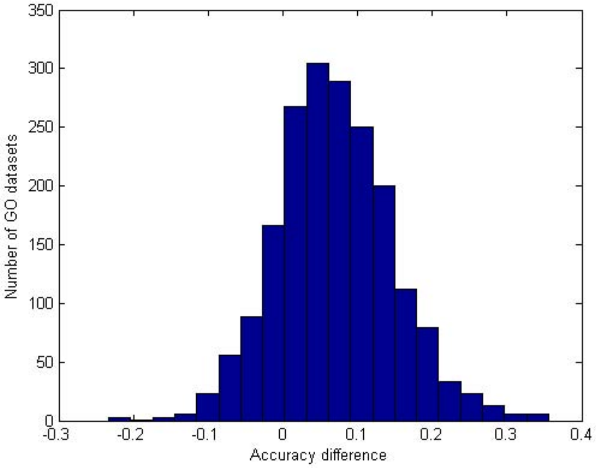


Fig. 2. Histogram of average accuracy differences between SVMs and CART decision trees obtained by 10 repetitions of 10 fold cross-validation over 1930 GO terms

CART decision trees we obtained 72.0%. Thus, on the majority of the GO terms SVMs perform better than CARTs.

We also performed experiments to reduce the dimensionality of the classification problems by Principal Component Analysis (PCA). During cross-validation trials we used balanced datasets with a number of principal components corresponding to the number of training examples. Note that for all the GO terms the dimensionality of the problem was always greater than the number of training proteins. The results are shown in Figure 3. For SVMs applied to the datasets representing the extracted principal components we obtained accuracy equal to 71.9%. Thus, on the majority of the GO terms SVMs perform better than SVMs applied to a reduced dimensionality problem. By increasing the proportion of negative proteins in the GO terms we found similar results.

We evaluated the one-versus-all classification strategy on the independent testing set of 44 proteins. We took into account 359 GO terms of the "F" subontology. For each GO term we combined 9 SVM classifiers for which we varied the proportion of negative proteins from 1 to 9. Note that with respect to the ninth classifiers the first classifier is more sensitive to positive proteins, but will also give more false positives.

For each protein of the testing set and for each GO term a score was calculated according to the proportion of classifiers that predicted the presence of each GO term. By sorting the scores of all the GO terms on a descending order we obtained a ranked list of predictions. The average recall on the first 50 ranked GO terms was 39/209 = 18.7% and the precision was 39/(50*44) = 1.8%. Cross-validation results emphasized much more optimistic recall/precision results. In other words, for the testing set too many false positives were present in the first positions of the predicted functions.

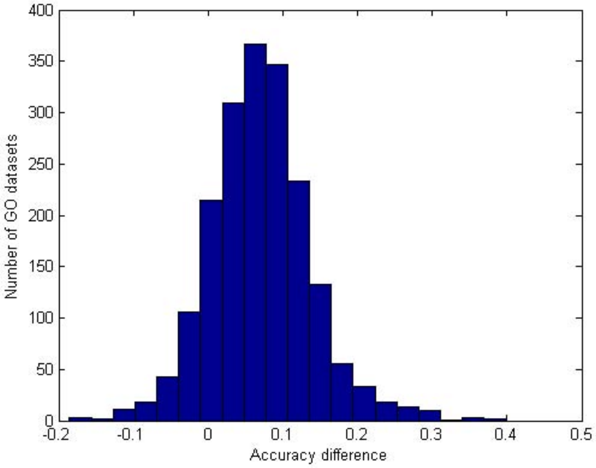


Fig. 3. Histogram of average accuracy differences between SVMs and SVMs applied to datasets with reduced dimensionality

4.2 Tournament Learning

In this series of experiments we focused on the 359 GO terms of the "F" branch. To apply a tournament learning strategy the key idea was to determine all the disjointed pairs of GO terms by examining whether proteins lie at the intersection. Then, each disjointed pair of GO terms can be learned by a binary classifier such as SVMs.

The average number of disjointed GO term classifiers was 259.4. Thus, for the tournament learning strategy we defined 46'564 binary classifiers corresponding to all the disjointed pairs. The union of the training proteins associated to the 359 GO terms represented more than 7'131 proteins. We defined the score of a predicted GO term as the proportion of classifiers predicting this GO term.

All the SVM predictors were applied to the independent dataset of 44 proteins to produce a ranked list of predicted functions. Figure 4 illustrates the histogram of matches with respect to their rank. With the first 50 ranked terms we had 108 matches, corresponding to average recall equal to 51.7% and average precision equal to 4.9%. Therefore, with respect to the previous experiments we improved the prediction performance.

The average rank is defined as the average of all the matched positions in the list of predicted functions. The average of the average ranks was 83.7.

4.3 Multi-label Learning

Again, we focused on the 359 GO terms of the "F" branch. Based on ten repetitions we first carried out 10-fold cross-validation trials on the proteins of the training set. We performed the computations with the Matlab software package proposed in [16]. The feed-forward neural architectures had 33'102 neurons in the

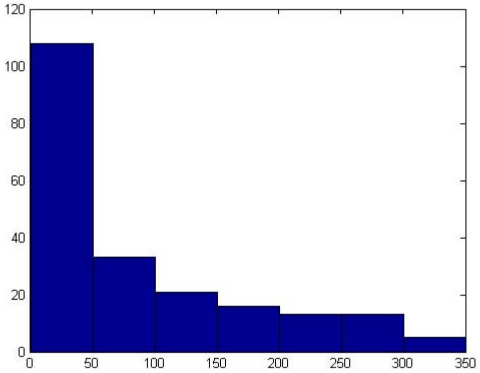


Fig. 4. Histogram of matched protein ranks for the tournament learning strategy

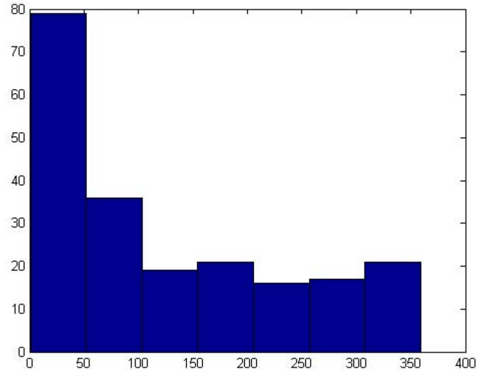


Fig. 5. Histogram of matched protein ranks for the multi-label learning strategy

input layer, 70 in the hidden layer and 359 neurons in the output layer. Learning parameters were set with the default values defined in the software package. During cross-validation trials we used a validation set to avoid the over-training phenomenon. The results were:

- Average precision = 63.4% (±0.5%)
- Average rank = 49.0 (±0.9)

Concerning the independent testing set, Figure 5 illustrates the histogram of matches with respect to their rank. For each protein with the first 50 ranked terms we had 79 matches, corresponding to average recall equal to 37.8% and average precision equal to 3.6%. Therefore, the prediction performance was worse than that obtained with the tournament learning strategy.

5 Discussion and Conclusion

We proposed to tackle the problem of human protein function prediction by three distinct supervised learning schemes : one-versus-all classification; tournament learning; multi-label learning (MLL). With respect to cross-validation trials performed on the training set the independent testing set was much more difficult to predict. The main reason probably resides in the fact that the training dataset encompasses a significant proportion of protein molecular functions that can be deduced directly from the presence of InterPro domains and PROSITE patterns/profiles.

We are not yet ready to predict difficult cases with high precision, as average precision on the difficult independent set was below 5%. However, on this dataset the average recall measure reached a reasonable performance for the first 50 ranked predictions, especially for the tournament learning scheme.

In the future we will try to improve the results by combining tournament learning and multi-label learning. Finally, rule extraction will also play an important role, as a major concern for Biologists is to understand how a machine learning model arrives at a particular prediction, especially when there is no explicit or simple relationship between inputs and outputs.

Acknowledgements

The authors gratefully thank SystemsX.ch for having funded this work under grant IPP 2009_41. All the computations were performed at the Vital-IT (http://www.vital-it.ch) Center for high-performance computing of the Swiss Institute of Bioinformatics.

References

1. Aerts, S., Lambrechts, D., Maity, S., Van Loo, P., Coessens, B., De Smet, F., Tranchevent, L.C., De Moor, B., Marynen, P., Hassan, B., et al.: Gene prioritization through genomic data fusion. Nat. Biotechnol. 24(5), 537–544 (2006)
2. Barutcuoglu, Z., Schapire, R.E., Troyanskaya, O.G.: Hierarchical multi-label prediction of gene function. Bioinformatics 22(7), 830–836 (2006)
3. Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J.: Classification and regression trees. Wadsworth, Belmont (1984)
4. Cai, C.Z., Han, L.Y., Ji, Z.L., Chen, X., Chen, Y.Z.: SVM-Prot: web based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Research 31(13), 3692–3697 (2003)
5. Eisenberg, D., Schwarz, E., Komaromy, M., Wall, R.: Analysis of membrane and surface protein sequences with the hydrophobic moment plot. J. Mol. Biol. 179(1), 125–142 (1984)
6. Hu, L., Huang, T., Shi, X., Lu, W.C., Cai, Y.D., Chou, K.C.: Predicting functions of proteins in mouse based on weighted protein-protein interaction network and protein hybrid properties. PLoS One 6(1), e14556 (2011)

7. Jensen, L.J., Gupta, R., Staerfeldt, H.-H., Brunak, S.: Prediction of human protein function according to gene ontology categories. Bioinformatics 19(5), 635–642 (2003)
8. Kazawa, H., Izumitani, T., Taira, H., Maeda, E.: Maximal margin labelling for multi-topic text categorization. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 17, pp. 649–656. MIT Press, Cambridge (2005)
9. Mewes, H.W., Heumann, K., Kaps, A., Mayer, K., Pfeiffer, F., Stocker, S., Frishman, D.: MIPS: a database for protein sequences and complete genomes. Nucl. Acids Research 27, 44–48 (1999)
10. Pena-Castillo, L., Tasan, M., Myers, C.L., Lee, H., Joshi, T., Zhang, C., Guan, Y., Leone, M., Pagnani, A., Kim, W.K., et al.: A critical assessment of Mus musculus gene function prediction using integrated genomic evidence. Genome Biol. 9(suppl. 1), S2 (2008)
11. Ranea, J.A., Yeats, C., Grant, A., Orengo, C.A.: Predicting protein function with hierarchical phylogenetic profiles: the Gene3D Phylo-Tuner method applied to eukaryotic genomes. PLoS Comput. Biol. 3(11), e237 (2007)
12. The Gene Ontology Consortium. The gene ontology project in 2008. Nucleic Acid Research 36(1), D440–D444 (November 2007)
13. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. International Journal of Data Warehouse and Mining 3(3), 1–13 (2007)
14. Vapnik, V.: The nature of statistical learning. Springer, Heidelberg (1995)
15. Vens, C., Struyf, J., Schietgat, L., Dzeroski, S.: Decision trees for hierarchical multi-label classification. Machine Learning 73(2), 185–214 (2008)
16. Zhang, M.L., Zhou, Z.H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering 18(10), 1338–1351 (2006)
17. Zhu, M., Gao, L., Guo, Z., Li, Y., Wang, D., Wang, J., Wang, C.: Globally predicting protein functions based on co-expressed protein-protein interaction networks and ontology taxonomy similarities. Gene 391(1-2), 113–119 (2007)

Evaluating Case Selection Algorithms for Analogical Reasoning Systems*

Eduardo Lupiani, Jose M. Juarez, Fernando Jimenez, and Jose Palma

Computer Science Faculty – Universidad de Murcia – Spain
{elupiani,jmjuarez,fernan,jtpalma}@um.es

Abstract. An essential issue for developing analogical reasoning systems (such as Case-Based Reasoning systems) is to build the case memory by selecting registers from an external database. This issue is called case selection and the literature provides a wealth of algorithms to deal with it. For any particular domain, to choose the case selection algorithm is a critical decision on the system design. Despite some algorithms obtain good results, a specific algorithms evaluation is needed. Most of the efforts done in this line focus on the number of registers selected and providing a simple evaluation of the system obtained. In some domains, however, the system must fulfil certain constraints related to accuracy or efficiency. For instance, in the medical field, specificity and sensitivity are critical values for some tests. In order to partially solve this problem, we propose an evaluation methodology to obtain the best case selection method for a given memory case. In order to demonstrate the usefulness of this methodology, we present new case selection algorithms based on evolutionary multi-objective optimization. We compare the classical algorithms and the multi-objective approach in order to select the most suitable case selection algorithm according to different standard problems.

1 Introduction

The cognitive process of reasoning by analogy comprehends a sort of different tasks such as obtaining abstract knowledge from instances or to solve new problems based on previous experiences [32]. One the one hand, the abstraction issue is studied, from a computational point of view, in the instance-based learning algorithms [3] or in more specific problems, such as temporal abstraction. On the other hand, analogical reasoning systems (ARSs) focus on solving a problem by comparing prior problem-solving episodes [8,12], stored in a case memory.

Over the years, ARS have become a well known field and have demonstrated to be essential in domains such as industry [20] or medicine [18]. Some of the most extended approaches are the K-NN family of algorithms and the Case-based Reasoning (CBR), a methodology to develop ARSs by providing a 4-step guided cycle [28], called CBR cycle [1].

A solid building process and a good evaluation of the knowledge base are essential in the practical application of ARS. Building processes mainly focus on defining (1) how

* This study was partially financed by the Spanish MEC through projects TIN2009-14372-C03-01 and PET2007-0033 and the regional project SENECA 15277/PI/10.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 344–353, 2011.

the cases (instances) are stored in the case memory, (2) a retrieval strategy to obtain the most similar cases from the case memory and (3) how the solution is built from these knowledge [13]. The evaluation of knowledge-based systems, including ARSs, depends on their knowledge base. Unlike other approaches, such as rule-based or model-based systems [19], in an ARS each piece of the knowledge base (cases of the case memory) is knowledge-complete and independent from the rest. Therefore, the main issue to build a case memory is to select which cases must be included or removed.

Two main approaches can be considered. Firstly, if the system is already established, the case memory must be *maintained*. This CBR task deals with the decision of removing, adding or modifying elements of the case memory [15]. Secondly, during the building process, the case memory can be obtained by selecting registers from an external data base. The automatic process to carry out this selection is known as instance selection, case selection or case mining [17,21].

The literature provides a wealth of case selection algorithms [9,7,2,22,27,29,31], however during the ARS building process, the selection of the most suitable algorithm is critical. Despite the impact of this process in the ARS evaluation, a few efforts have been done in this direction.

This work focuses on the case selection task and its evaluation. To this end, we introduce an evaluation method for case selection algorithms. In order to demonstrate the usefulness of this methodology, we present new case selection algorithms based on multi-objective evolutionary optimization. We compare the classical algorithms and the multi-objective approach in order to select the most suitable case selection algorithm according to different standard problems.

The structure of this work is as follows. Section 2 provides a review of the related works on case selection methods and evaluation. In Section 3 we propose a evaluation methodology for case selection methods. Section 4 describes the experiments and results. Finally, conclusions and future works are described in Section 5.

2 Related Works

The present work is aimed to evaluate case selection methods. We briefly review what kind of case selection families and which evaluation techniques have been studied in literature.

Case selection methods can be classified either by the case selection methodology [30,31] or by case memory construction technique [21]. Among the case selection methods there are four outstanding families: (i) nearest neighbour editing rules [9,7,29,22,27], (ii) instance-based [3,2,31], (iii) prototype-based [5] and (iv) competence-based [24,26,17]. Both (i) and (ii) select a case as candidate to final case memory in basis to K-NN classification and (iii) could additionally adapt cases or modify them. Specially interesting methods are those based on competence techniques (iv) since they introduce new concepts such as coverage and reachability; however they need good understanding of domain problem. Case selection and feature selection approaches has been very popular research issue to reduce noise in case memory and enhance system response times respectively, however almost all studies face them separately. Multi-objetive evolutionary algorithms (MOEA) makes possible to cope both objectives at the same time [14,4]. Nevertheless, most experiments fall into evolutionary algorithms test sets and they are not based in popular MOEA implementations such as NSGA-II [6] and SPEA2 [33].

The Hold-Out validation is a suitable approach to evaluate the case selection methods due to large size of initial data base. Experiments as [22,21,25,23,7,16] uses Hold-Out repeatedly a determined number of times. Furthermore, some studies try to analyse the effects of case memory size in the classifier accuracy. However, this studies are restricted to K-NN based classifiers. In this cases, the case selection process finishes when a concrete (and pre-established) size of the case memory is reached [21,25,16]. Although Hold-Out is appropriate for large cases sets, it could present a high variance. Therefore, some researchers prefer Cross-Validation [31].

3 Evaluation Methodology and Case Selection Methods

3.1 Notation

Let C be the universe of all possible cases in a give domain and $c \in C$ be a particular case. We assume that a case is a vector of n attributes $x = (x_1, \cdots, x_n)$, where each attribute could describe a quantitative or qualitative value.

Let also define $\mathbb{M} = \wp(C)$ as the space of possible case memories and $M \in \mathbb{M}$ a particular case memory. Therefore, a case c of M is denoted by $c \in M$.

Let $\{M_i, i \in \{1, \cdots, f\}\}$ be a partition of M ($\bigcup_{i=1}^{f} M_i = M, \forall i, j M_i \cap M_j = \emptyset$). We define the complement of M_i (element of a partition of M) as $\overline{M_i} = M \setminus M_i$.

A case selection method can be defined by the function σ as follows:

$$\sigma : \mathbb{M} \rightarrow \mathbb{M} \tag{1}$$

where $|\sigma(M)| \leq |M|$. For the sake of clarity, we denote M^σ as $\sigma(M)$. In Section 2, the implementation of σ functions proposed in the literature are briefly reviewed and Section 3.3 describes in depth some of them.

Given M^σ we define the reduction rate as:

$$\rho : \mathbb{M}^\sigma \rightarrow \{0, 1\} \tag{2}$$

$$\rho(M^\sigma) = \frac{|M| - |M^\sigma|}{|M|}. \tag{3}$$

Therefore, $\rho(M^\sigma) = 0$ means that σ makes no reduction, and $\rho(M^\sigma) = 1$ implies that no cases were selected from M using the σ function.

3.2 Evaluation Methodology

The final goal of the evaluation methodology is to obtain the best case selection method for a given memory case for solving a particular problem. According to this vague definition, we assume 3 main dimensions: the size of the case memory after the selection process, the efficiency of the method, and the suitability of the case memory to solve the problem.

In order to analyse these aspects we propose a 4-step methodology to evaluate case selection algorithms:

1. Calculate $\{M_i, i \in \{1 \cdots f\}\}$, a randomly partition of M and $\overline{M_i}$ the complement of each element M_i.
2. For each $\overline{M_i}$ apply the control and the case selection method ($\overline{M_i}^{\sigma}$).
3. Validate the classifier using Cross-Validation where $\overline{M_i}^{\sigma}$ is the training set and M_i the test set.
4. Calculate the decision scores: reduction of the case memory, efficiency of the method, and quality of the solution.

In the first step, the partition is made to identify the test and training sets. In the second step, the case selection method is applied in order to reduce the case memory. Note that the case selection method selected should not produce an adverse effect on the ARS. In order to obtain an initial filtering of the case selection methods, we could compare them with a control test. In our case, these control methods are: the random selection process (removing 25,50 or 75 % of the cases from the case memory) and the *none* selection (keeping the original case memory). Therefore, this methodology only considers acceptable those case selection methods whose results improve or keep the control methods.

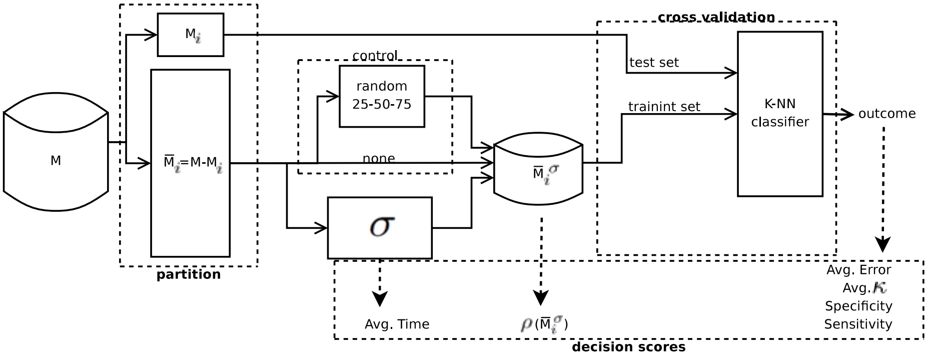


Fig. 1. Evaluation methodology for case selection algorithms

The third step is a classical Cross-Validation process. Due to the fact that case selection methods are used to improve ARSs, it seems reasonable to include the own ARS at this step. However, this kind of systems (such as a CBR) could imply high computationally-cost processes (e.g. similarity or adaptation functions) and the validation step implies a high number of iterations. Therefore, the custom cross-validation presented (folder size f) executes a case selection method using the training set $\overline{M_i}^{\sigma}$, the test set M_i, and the K-NN as classifier (iterating over $i = 1, \cdots, f$). The K-NN has two components: local and global distances, where the global depends on local. The local is the distance between the case attributes values, therefore its calculation depends on the attribute type. In our evaluation there are just two types: numeric and string of characters, and we call d_{num} the distance between numeric values and d_{string} the distance between string values:

$$d_{num}(x_i, y_i) = \frac{|x_i - y_i|}{|max(x_i) - max(y_i)|} \quad (4)$$

$$d_{string}(x_i, y_i) = \begin{cases} 1 \text{ if } x_i \neq y_i \\ 0 \text{ if } x_i = y_i \end{cases} \quad (5)$$

where $max(x_i)$ and $max(y_i)$ are the maximum domain values for the attributes x_i and y_i.

Given two particular cases $x = (x_1, \cdots, x_n)$ and $y = (y_1, \ldots, y_n)$ with the same number of attributes, the global euclidean distance is defined as:

$$D(x, y) = \sqrt{\sum_{i=1}^{n} d_i(x_i, y_i)^2}. \quad (6)$$

where d_i is d_{num} or d_{string} depending on the nature of the *i-th* attribute.

In the final step of the methodology proposed the reduction of the case memory is evaluated by the reduction rate function ρ (expression 2). The efficiency is calculated by the average of the execution time of the reduction process. Finally, we consider the generalised error rate to evaluate the classification, the κ coefficient to evaluate the coincidence of the solution and specificity and sensitivity values to analyse particular problems of the domain application.

3.3 Case Selection Methods

In this work, we analyse three main families of case selection algorithms: NN algorighms, instance-based algorithms, and drop algorithms. Since the evaluation criteria of the case selection methods are their reduction capacity and the accuracy of the classifier, we could consider it as a multiobjective problem. Therefore, we propose the use of evolutive algorithms to implement case selection.

NN Family. One of the first attempts to reduce case memory size was CNN [9]. Starting with a random case of each class as final case memory, the method takes the rest of them as test set and classifies them using the selected cases as K-NN classifier. If a case is misclassified then it is added to the final case memory. The process stops when all the original cases are right classified. Although the reduction is possible, nevertheless it does not check for noisy cases, for this reason RNN [7] was introduced as an extension of CNN to remove noise instances from the resulting case memory after applying CNN. Each case of the final case memory is removed from it, and if no case from the original memory is misclassified then the candidate is finally removed, in other case it is kept.

Unlike RNN, ENN does not start with the output memory of CNN. ENN removes misclassified cases by their three nearest neighbours [29]. Consequently, if ENN is executed multiple times, taking each output as input of the next execution , then the method is called RENN. All-KNN consists of executing k times ENN, where each execution uses from 1 to k neighbours respectively to flag a case for no selection [27]. Some authors claim ENN and its variations are actually noise remove techniques [31].

Instance-based Family. In IB2 if an instance is misclassified by its nearest neighbour then it is added to the final case [3,2]. Like CNN, the resulting memory could be highly noisy. IB3 sets a more restrictive condition to keep a selected instances inside the final case memory by reducing noise presence: if the instance in the memory has a low acceptability level, i.e. low accuracy, then it is removed.

Shrink method [11], as RNN extends CNN. Shrink executes CNN and, finally, it removes the instances missclassified from the resulting case memory by theirs neighbours.

DROP Family. The reduction methods DROP1, DROP2 and DROP3 were introduced as methodologies that provide noise tolerance, high generalisations accuracy, insensitivity to the order of presentation of instances, and significant storage reduction [31]. All this method introduce the concept of associate, which it is the nearest neighbours to a case of the same class. DROP1 is identical to RNN with the exception that the accuracy is checked on the selected instances compounding the final memory case, where initially all the cases are by default selected. A case is deselected if at least as many of its associates, previously selected, is classified correctly without it. The difference with DROP2 is that the case is deselected if at least as many of its associates in the original memory case would be classified correctly without it. DROP3 just executes ENN before DROP1.

An Evolutionary Multiobjective Optimization Approach for Case Selection. The selection of cases concerns finding the smallest subset of cases in a data base to obtain the most accurate classification possible. Described more formally, lets suppose an initial case memory M where $|M| = X$, the algorithm finds $M^\sigma \subseteq M$, removing the irrelevant or redundant cases, and obtaining good accuracy of the classification. For the sake of clarity, since the algorithm could obtain different M^σ sets, we denote them by x, y, z, etc.

Therefore, as in [10] for attribute selection, the problem of cases selection can be approached as a multiobjective optimization problem, the solution of which comprise as set of solutions called non-dominated solutions (or Pareto solutions). Given two solutions $x = \{c|c \in M\}$ and $y = \{c|c \in M\}$, solution x dominates solution y if [6]:

- Solution x is not worse than y for any of the purposes in mind;
- Solution x is strictly better than y for at least one of the objectives.

For the case selection problem in mind, two optimization criteria have been considered: *accuracy* and *compactness*. To formulate these criteria the following quantitative measures have been defined.

Given a solution x of M, we define:

- Accuracy. Based on the error ratio $ER(x) = \dfrac{\Phi(x)}{|x|}$, where $\Phi(x)$ is the number of cases misclassified for a set of cases, x, by a given classification algorithm.
- Compactness. By cardinality $|x|$, that is, the number of cases used to construct the model.

According to these criteria, we propose the following optimization model:

$$\begin{aligned} &Minimize\ ER(x) \\ &Minimize\ |x| \end{aligned} \tag{7}$$

Note that the objectives in the optimization model (7) are contradictory since a lower number of significant cases means a higher error rate and vice versa, that is the greater the number of variables the smaller the error rate. The solution to model (7) is a set of $m \leq X$ non-dominated solutions $C = \{x^k, k \in S\}$, $S = \{1, \ldots, X\}$, where each solution x^k of C represents the best collection of significant k cases.

From the practical point of view and in order to simplify the model, it is interesting to sacrifice accuracy slightly when the number of cases are reduced significantly. In Section 4 some examples are provided.

We propose NSGA-II [6] and SPEA-2 [33] Multiobjective Evolutionary Algorithms to solve the problem.

4 Experiments and Results

In this section, we present a practical use of the methodology proposed. We evaluate the case selection methods described in Section 3.3 using case memories of different domains. In particular we consider standard datasets of the UCI repository[1]. Following this methodology, we set $f = 10$ (that is, Cross-Validation with 10 folders) and $K = 1$ (i.e. 1-NN classifier).

Table 1 depicts a summary of the experiments. For each case memory (rows), the best results are enhanced in boldface and the worst in italics.

In general, CNN and RNN achieve great size reductions in noise free case memories, however they keep the error higher than the control methods in every case. In all the experiments, ENN and All-KNN maintain or improve the error rate. If the case memory has no noisy instances then the reduction is negligible, otherwise the reduction is clearly significant. Method as IB2, IB3 and Shrink achieve great size reductions when they select instances from a case memory with well defined boundaries, but they are too weak to presence of noisy instance.

The evaluation of methods considering each dataset highlights the suitability of some methods. For small datasets with high defined boundaries (Iris and Wine), SPEA-2 algorithm seems to be the best approach since they reduce about 50% of the case memory maintaining an acceptable error rate. For larger datasets and no clear boundaries (Yeast or Breast Cancer datasets), DROP algorithms also obtain an effective reduction of de memory (approx. 80%), however ENN and RENN reach a solid reduction but minimises the error rate with a less time cost. Note that in the medical domain (e.g. Breast Cancer dataset) other aspects must be taken into account. In this sense, NSGA-II seems to be the most effective algorithm since reduces about 50% the case memory maintaining the error rate, but also maintaing the kappa coefficient, specificity and sensitivity. According to the experiments, ENN and RENN seem also useful for large datasets with a high number of classes (such as Abalone), improving the accuracy of the system by reducing 80% the case memory.

[1] http://archive.ics.uci.edu/ml/

Table 1. Decision score results of the experiments: err. is the error rate, ρ is the reduction rate, time (in seconds), Sens is the sensitivity, Spec. is the specificity and κ is the Kappa coefficient.

	Iris		Wine		Yeast			Breast Cancer						Abalone		
	Err.	ρ	Err.	ρ	Err.	ρ	time	Err.	ρ	time	Sens.	Spec.	κ	Err.	ρ	time
None	0,05	0	0,03	0	0,48	0	0	0,28	0	0	0,49	0,82	0,3	0,8	0	0
Rand. 25%	0,04	0,25	0,04	0,25	0,5	0,25	0,02	0,3	0,25	0	0,42	0,81	0,24	0,79	0,25	0,18
Rand. 50%	0,05	0,5	0,06	0,5	0,52	0,5	0,05	0,33	0,5	0	0,44	0,77	0,2	0,79	0,5	0,38
Rand. 75%	0,07	0,75	0,07	0,75	0,5	0,75	0,08	0,33	0,75	0	0,33	0,81	0,15	0,8	0,75	0,57
CNN	0,08	0,85	0,08	0,89	0,52	0,33	0,05	0,33	0,46	0	0,44	0,77	0,21	0,81	0,11	1,26
RNN	0,07	0,85	0,06	0,85	0,52	0,33	0,06	0,34	0,46	0	0,42	0,76	0,18	0,81	0,11	2,91
ENN	**0,05**	*0,04*	0,04	*0,04*	**0,42**	0,49	0,01	**0,24**	*0,24*	0	0,29	0,94	0,24	**0,74**	0,83	0,02
RENN	**0,05**	*0,04*	0,04	*0,04*	**0,42**	0,49	0,03	**0,24**	*0,24*	0	0,29	0,94	0,24	**0,74**	0,83	0,08
All-KNN	0,06	0,09	**0,03**	0,05	0,45	0,57	0,01	0,28	0,4	0	0,36	0,87	0,23	0,77	0,85	0,05
IB2	0,16	**0,95**	0,12	**0,96**	*0,53*	0,44	0,01	0,37	0,61	0	0,47	0,69	0,16	0,81	0,19	0,01
IB3	*0,17*	0,88	*0,14*	0,82	*0,53*	0,44	*139,09*	*0,39*	0,56	9,05	0,44	0,68	0,12	0,81	0,19	*1132,19*
Shrink	0,09	0,9	0,11	0,91	*0,53*	0,42	0,06	0,37	0,58	0	0,41	0,72	0,13	0,81	0,16	1,67
DROP1	0,05	0,94	0,07	0,9	0,48	0,76	0,45	0,26	**0,83**	0,04	0,38	0,89	0,28	0,78	0,83	2,8
DROP2	0,05	0,94	0,07	0,87	0,49	**0,81**	0,49	0,26	0,81	0,04	0,37	0,89	0,26	0,78	**0,89**	3,01
DROP3	0,05	0,88	0,07	0,82	0,52	*0,32*	0,2	0,34	0,5	0,02	0,51	0,72	0,23	*0,82*	*0,1*	0,46
SPEA2	0,05	0,57	0,05	0,54	0,5	0,51	37,33	0,31	0,52	8,2	0,44	0,79	0,22	0,8	0,5	183,15
NSGA-II	0,06	0,5	0,05	0,51	0,51	0,5	42,48	0,28	0,51	*9,32*	0,47	0,82	0,3	0,8	0,5	204,14

It is worht mentioning that the MOEA algorithms have similar behaviour and achieve similar results in the reduction rate and the error. Their reductions are near to the half size of case memory in all cases, and the results are a slightly higher than control experiments. They show the same result in every experiment so although they are not the best in any case, they are the most regular respecting the instances distribution. Their weakness is the execution time expended of them. The rest of methods are highly efficient respect to them.

5 Conclusions

The aim of this work is to provide support to choose the most suitable case selection method for a given memory case to solve a particular problem. To this end, we propose an evaluation methodology based on the size of the case memory after the selection process, the efficiency of the method, and the suitability of the case memory to solve the problem.

In some works, the evaluation of the case selection process focuses on the validation of the complete system using an arbitrary repetition of the Hold-Out technique [22,23,7]. Our methodology proposes the use of a K-NN classifier, avoiding the use of the complete system. The works described in [21,25,16] also suggest the same strategy but the case selection process is stopped when the final case memory reaches a determinate size. Unlike our methodology, the case memory size must be known beforehand and this assumption could not be acceptable in some domains. Moreover we suggest the use of a Coss-Validation, providing a more flexible approach by tuning the folder size.

The experiments carried out show in different standard problems provide a clear proof of the potential application of the methodology proposed. Future works will focus on the application of our methodology in order to develop case-based reasoning systems in real medical applications.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 7, 39–59 (1994)
2. Aha, D.W.: Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. International Journal Of Man-Machine Studies 36, 267–287 (1992)
3. Aha, D.W., Kiblerand, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning 6, 37–66 (1991)
4. Ahn, H., Kim, K., Han, I.: A case-based reasoning system with the two-dimensional reduction technique for customer classification. Expert Systems With Applications 32, 1011–1019 (2007)
5. Chang, C.L.: Finding prototypes for nearest neighbor classifiers. IEEE Transactions on Computers C23, 1179–1184 (1974)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
7. Gates, G.W.: The reduced nearest neighbor rule. IEEE Transactions On Information Theory 18, 431–433 (1972)
8. Hall, R.P.: Computational approaches to analogical reasoning. Artificial Intelligence 39, 39–120 (1989)
9. Hart, P.E.: The condensed nearest neighbor rule. IEEE Transaction On Information Theory 14, 515–516 (1968)
10. Jara, A., Martinez, R., Vigueras, D., Sanchez, G., Jimenez, F.: Attribute selection by multiobjective evolutionary computation applied to mortality from infection severe burns patients. In: Proceedings of the International Conference of Health Informatics (HEALTHINF 2011), Algarbe, Portugal, pp. 467–471 (2011)
11. Kibler, D., Aha, D.W.: Learning representative exemplars of concepts: An initial case study. In: Proceedings of the Fourth International Workshop on Machine Learning, pp. 24–30 (1987)
12. Kling, R.E.: A paradigm for reasoning by analogy. Artificial Intelligence (2),147–148 (1971)
13. Kolodner, J.L.: Making the Implicit Explicit: Clarifying the Principles of Case-Based Reasoning. In: Case-based Reasoning: Experiences, Lessons and Future Directions, ch. 16, pp. 349–370. AAAI, Menlo Park (1996)
14. Kuncheva, L.I., Jain, L.C.: Nearest neighbor classifier: Simultaneous editing and feature selection. Pattern Recognition Letters 20, 1149–1156 (1999)
15. Leake, D.B., Wilson, D.C.: Categorizing case-base maintenance: Dimensions and directions. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS (LNAI), vol. 1488, pp. 196–207. Springer, Heidelberg (1998)
16. McKenna, E., Smyth, B.: Competence-guided case-base editing techniques. In: Blanzieri, E., Portinale, L. (eds.) EWCBR 2000. LNCS (LNAI), vol. 1898, pp. 186–197. Springer, Heidelberg (2000)
17. McSherry, D.: Automating case selection in the construction of a case library. Knowledge-Based Systems 13, 133–140 (2000)

18. Montani, S., Portinale, L., Leonardi, G., Bellazzi, R., Bellazzi, R.: Case-based retrieval to support the treatment of end stage renal failure patients. Artificial Intelligence in Medicine 37(1), 31–42 (2006)
19. Nersessian, N.: The cognitive basis of model-based reasoning in science. In: The Cognitive Basis of Science, ch. 7, pp. 133–153. Cambridge University Press, Cambridge (2002)
20. Olsson, E., Funk, P., Xiong, N.: Fault diagnosis in industry using sensor readings and case-based reasoning. Journal of Intelligent and Fuzzy Systems 15(1), 41–46 (2004)
21. Pan, R., Yang, Q., Pan, S.J.: Mining competent case bases for case-based reasoning. Artificial Intelligence 171, 1039–1068 (2007)
22. Ritter, G.L., Woodruff, H.B., Lowry, S.R., Isenhour, T.L.: Algorithm for a selective nearest neighbor decision rule. IEEE Transactions on Information Theory 21, 665–669 (1975)
23. Smyth, B., Keane, M.T.: Remembering to forget - A competence-preserving case deletion policy for case-based reasoning systems. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 1995), August 20-25, vol. 1 and 2, pp. 377–382 (1995)
24. Smyth, B., McKenna, E.: Modelling the competence of case-bases. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS (LNAI), vol. 1488, pp. 208–220. Springer, Heidelberg (1998)
25. Smyth, B., McKenna, E.: Building compact competent case-bases. In: Althoff, K.-D., Bergmann, R., Branting, L.K. (eds.) ICCBR 1999. LNCS (LNAI), vol. 1650, pp. 329–342. Springer, Heidelberg (1999)
26. Smyth, B., McKenna, E.: Competence models and the maintenance problem. Computational Intelligence 17, 235–249 (2001)
27. Tomek, I.: An experiment with the edited nearest-neighbor rule. IEEE Transactions On Systems Man and Cybernetics 6, 448–452 (1976)
28. Watson, I.: Case-based reasoning is a methodology not a technology. Knowledge-Based Systems 12, 303–308 (1999)
29. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions On Systems Man and Cybernetics SMC 2, 408–420 (1972)
30. Wilson, D.R., Martinez, T.R.: Instance pruning techniques. In: Machine Learning: Proceedings of the Fourteenth International Conference (ICML 1997), pp. 404–411. Morgan Kaufmann, San Francisco (1997)
31. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning 38, 257–286 (2000)
32. Winston, P.H.: Learning and reasoning by analogy. Commun. ACM 23, 689–703 (1980)
33. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, Athens, Greece, pp. 95–100. International Center for Numerical Methods in Engineering (2001)

On the Use of Human-Guided Evolutionary Algorithms for Tackling 2D Packing Problems

Javier Espinar, Carlos Cotta, and Antonio J. Fernández Leiva

Dept. Lenguajes y Ciencias de la Computación, ETSI Informática,
Campus de Teatinos, Universidad de Málaga,
29071 Málaga – Spain
{ccottap,afdez}@lcc.uma.es

Abstract. We consider a 2D packing problem in which a collection of rectangular objects have to be arranged within a larger rectangular area of fixed width, such that its height is minimized. This problem is tackled using evolutionary algorithms that combine permutational decoders and GRASP-based principles. It is shown that this approach can be improved by allowing the user interact with the algorithm, tuning the greediness of the genotype-to-phenotype decoding. Experiments are presented on three different problem instances with sizes ranging from 19 up to 49 objects.

1 Introduction

Cutting and packing (C&P) problems are common in many industrial applications. Albeit under different formulations, this kind of problems share the need of finding a dense arrangement of a collection of objects, in order to minimize the material needed for producing them or the cost of wrapping and shipping them, just to cite a couple of examples [1]. Problems in this area range from 1D C&P (e.g., cutting large wooden boards into shorter pieces) to 3D C&P (e.g., container loading), including mixed versions thereof. We specifically consider 2D C&P problems in this work, and more precisely the case in which objects have rectangular shape. Such problems naturally arise in manufacturing processes of wood, glass and paper industries [2].

In general problems in this area cannot be efficiently solved with complete algorithms (in fact, note that several classical problems such as the knapsack problem or vector packing [3] can be formulated as C&P problems, and hence they are in general NP-hard). For this reason, heuristic and metaheuristics have been often used to solve this kind of problems – see [4] for a review. In this sense, it is well-known that in order to attain effective (meta)heuristic solvers, problem knowledge must be exploited [5,6]. Optimization tools can be endowed of this problem-knowledge in many ways: ad hoc representations, specialized operators, or combination with other problem-specific techniques, just to cite a few [7]. We present an algorithmic approach that incorporates some commonly-used heuristic in the area, driven by GRASP-like procedure [8]. Our main focus is however

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 354–361, 2011.

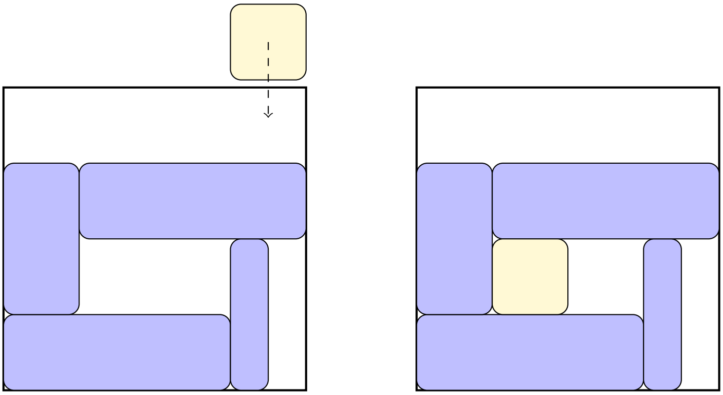


Fig. 1. Example application of the Bottom-Left-Fill heuristic. The square does not fit in the bottommost space at the right end of the figure, but does fit in the next empty space in a bottom-left fashion. The right figure shows the arrangement after the square has been placed.

in the use of a human expert to oversee the optimization process. This kind of application is typically termed interactive evolutionary computation (IEC) [9,10]. We prefer the term user-centric evolutionary computation [11] though, since it captures not just interactivity but also the possibility for proactivity. Be as it may, in this work we consider a human-guided approach in which the user asynchronously interacts with the algorithm in order to tune the greediness of the genotype-to-phenotype mapping, and use it as a proof of concept of the applicability of this kind of approaches in this problem domain.

2 Problem Definition and Background

Let $c = \{o_1, \cdots, o_n\}$ be a collection of n objects, each of them characterized by its width w_i and height h_i. We have to arrange the objects within a larger rectangle of fixed width W, such that (1) no two objects overlap, (2) they fit within the width of the larger container, and (3) the height of the arrangement (largest distance between the base and top of two objects) is minimized. When creating the arrangement, objects can be rotated by 90°, and no guillotine-constraint exists [1]. The so-defined problem naturally captures many manufacturing processes in wood, glass, and paper industries, and even problems from other domains such as processor allocation [12].

There have been numerous attempts at solving this kind of problem with metaheuristics. Burke and Kendall [13] compared tabu search, simulated annealing and genetic algorithms, and concluded the latter was outperformed by the former. Indeed, cutting and packing problems are known to pose difficulties to evolutionary approaches if these are unable to deal adequately with the relevant information pieces contained in solutions. Grouping genetic algorithms

[14] are actually based on this fact, and attempt to process the group structure of solutions. On the other hand, another very common feature of evolutionary approaches to this problem is trying to incorporate some kind of placement heuristic. For example, in the Bottom-Left-Fill (BLF) heuristic [15] objects are suitably ordered and then placed in the bottommost leftmost position (see Fig. 1). Other approaches such as Best-Fit (BF) [16] try to select the best object (if any) to be placed in each position, starting from the bottommost leftmost one. BLF heuristics are very well suited to combinations with metaheuristic approaches finding adequate orderings of the objects to be fed to the the placement heuristic. This has been for example the approach considered in [2] in the context of genetic algorithms, simulated annealing and hill climbing. As to the BF heuristic, it has been used by [17] in the context of GRASP and variable neighborhood search. As show in next section, the approach presented in this work combines the BLF heuristic with an evolutionary GRASP procedure, further enhanced by human intervention.

3 A Human-Guided Hybrid Evolutionary Approach

As anticipated in previous sections, there are two main ingredients in our evolutionary approach: a purely algorithmic component based on an underlying placement heuristic and an evolutionarily-driven greedy decoder, and a human-guided aspect. Let us describe these two elements separately.

3.1 The Hybrid Algorithm

The metaheuristic algorithm (which we will refer to as 'automated' henceforth, to distinguish it from its human-guided counterpart), combines ideas of permutational decoders and GRASP-based decoders. Essentially, we consider the two key elements that drive the functioning of a basic placement heuristic, namely the ordering of objects and the ordering of free slots where they can be accommodated. The permutational aspect is similar to the approach used in [2], that is, an evolutionary algorithm (EA) is used to evolve good object-orderings to be fed to a placement heuristic (BLF in our case). If this were the unique component of the algorithm, the EA would manage a population of solutions, each of which would be a permutation of elements $\{1, \cdots, n\}$. However, this representation is unable to capture the possibility of object rotations (or at least deprives the EA of control over this feature, that would be then hardwired within the placement heuristic). A simple way to allow direct evolutionary control over this might be to augment chromosomes with a n-bit string indicating the orientation of each object. We have however opted for a more general approach based on GRASP-like decoders.

To understand this decoder, let us firstly recall that GRASP (greedy randomized adaptive search procedures [18]) are construction techniques that build solutions incrementally, using a greedy heuristic as guidance mechanism. More precisely greediness is tempered by not taking always the locally optimal decision, but by making random decisions within a certain short list of candidate decisions (e.g., the k best decisions for some k, fixed in advance or reactively learnt [19,17]; other procedures exist though). This random selection of candidates can be substituted by an evolutionary approach, as shown in [8] for the Golomb ruler problem. The idea here is to consider a n-integer string $\langle s_1, \cdots, s_n \rangle$, where $s_i > 0$; s_i would thus indicate that when taking the i-th decision, the s_i-th best candidate should be taken (notice therefore that $\langle 1, \cdots, 1 \rangle$ would correspond to the purely greedy solution).

The representation of solutions combine both aspects, i.e., a permutation of the objects $\langle p_1, \cdots, p_n \rangle$ and a greedy-control sequence $\langle s_1, \cdots, s_n \rangle$. Hence, the interpretation of a certain solution is therefore: pick object p_1 and place it in the s_1-th place where it fits, counting from bottom to top and from left to right; the object p_2 and so on. In order to count candidate places, objects are firstly tried in the original orientation, and then rotated by 90°. If the object fits in both ways, each orientation is given a different index. Therefore, the EA has control on which position an object is placed, and with which orientation.

3.2 Adding Human Guidance

The addition of human-guidance to the evolutionary optimization process leads to interactive evolutionary computation (IEC). IEC basically consists of a computational model that promotes the communication between a human user and an automated evolutionary algorithm (aEA). This intervention of the human user is required when some aspect of the algorithm cannot be algorithmically construed. A typical example is the use of a human expert to provide subjective fitness evaluation of candidate solutions. This does not exhaust the possibilities though; the user can also modify the genetic contents of the population, or change dynamically some aspect of the algorithm.

One of the most typical problems IEC has to deal with is the fatigue of the human user. The latter cannot be forced to continuously supply feedback, since otherwise she would be soon fatigued and the quality of the feedback would sharply decrease. To alleviate this fatigue, our approach conceives the interaction as a purely asynchronous feature, i.e., the human-guided EA (hgEA) behaves as an aEA and hence is capable of working without user intervention. However, it is also capable of being interrupted by the user who can interact with the algorithm and resume the search subsequently.

In our approach interaction can take many forms as anticipated above: the user can inspect statistical information of the population (frequencies of each allele, correlation of an allele with fitness, etc.), as well as individual solutions, and take decisions such as manually select solutions for breeding, and change operators and parameters, just to cite a few. In this particular application we have focused on the capability for dynamically changing the range of values each gene can take. More precisely we concentrate on the greedy-control part and let the user specify online the range of values these genes can take. By doing so, the user is capable of adjust the greediness of the genotype-to-phenotype mapping,

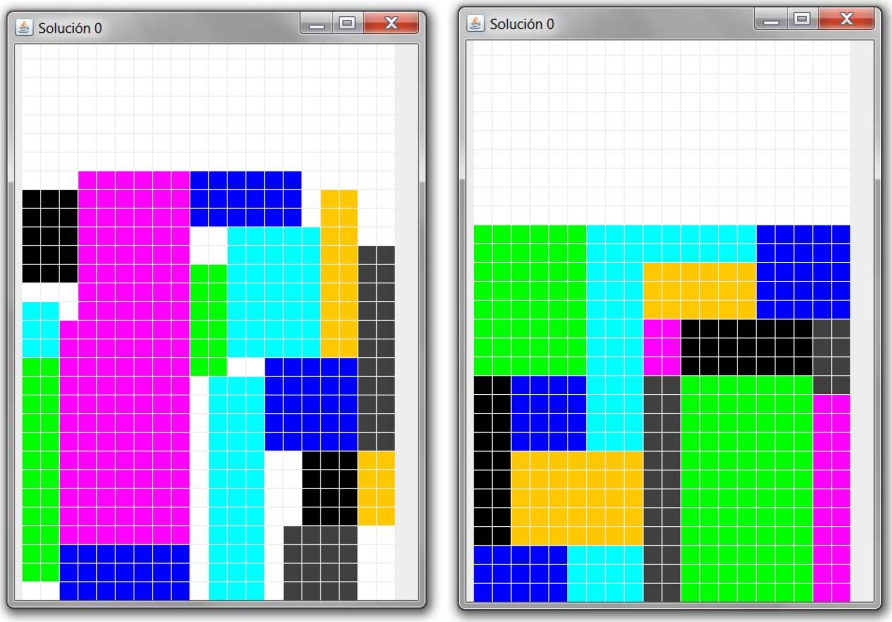


Fig. 2. Best solution found in the 16-object instance with the automated EA (left), and the human-guided EA (right)

forcing a highly-greedy decoding process by choosing a short range of values above zero for genes, or allowing more exploratory strategies by selecting a more ample value range for genes.

4 Experiments

The experiments have been done with an (μ, λ)-EA ($\mu = 100$, $\lambda = 90$, $p_X = 0.8$, $p_M = 0.1$). In order to encode the permutation part of the chromosome, a stack-based encoding [20] is used, i.e., a permutation $\langle p_1, \cdots, p_n \rangle$ is encoded as an integer list $\langle e_1, \cdots, e_{n-1} \rangle$ where $1 \leqslant e_i \leqslant (n - i + 1)$; here, a base ordering is assumed, say, $p_i = i$, and then e_i indicates the index of the element of this base ordering that is selected (without replacement) as the i-th element of the permutation. Notice that there is just one choice for the very last element, and hence the encoding only needs $n - 1$ values. Furthermore, this encoding is amenable for simple variation operators, since any sequence represents a valid permutation as long as each value e_i is within the required range. The same applies to the greedy control sequence, and therefore standard one-point crossover and random-substitution mutation have been used.

Three problem instances with $n = 16$, 29 and 49 objects have been selected from [2]. In every case the maximum number of evaluations is set to *maxevals*$=$ $1000n$. Twenty runs per problem instance are done. In the case of the hgEA,

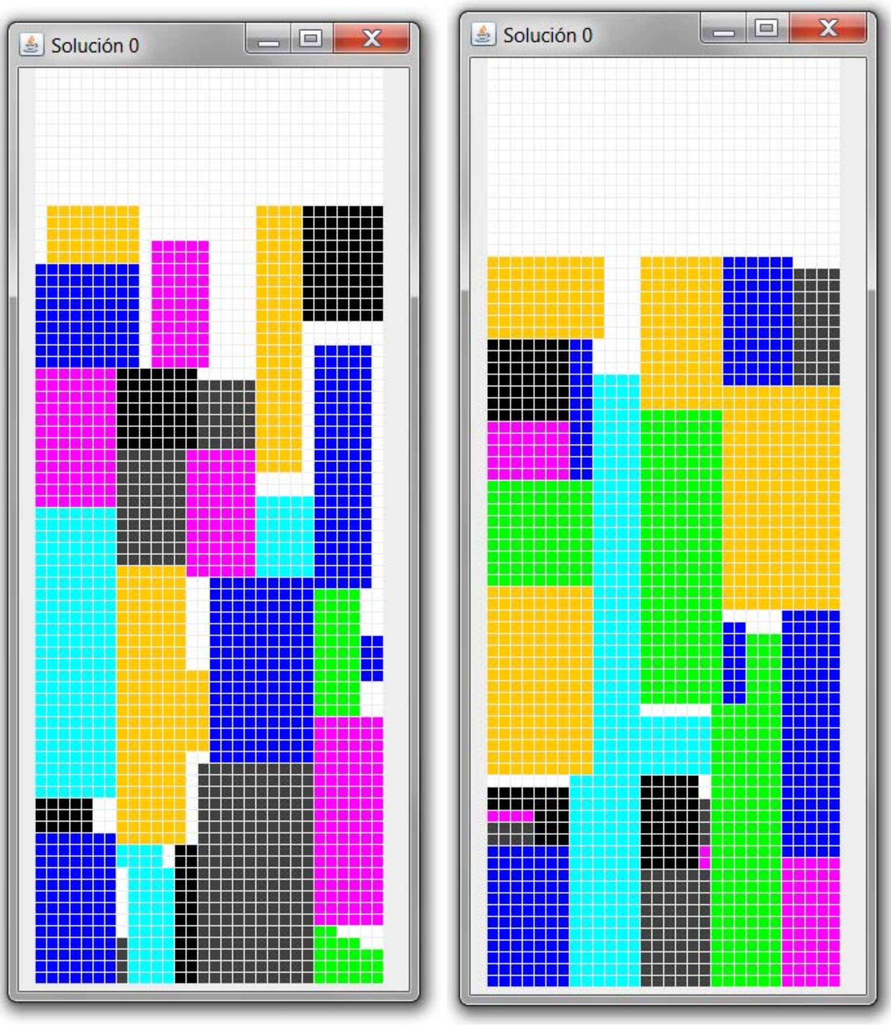


Fig. 3. Best solution found in the 29-object instance with the automated EA (left), and the human-guided EA (right)

the human interaction is focused on augmenting greediness in the initial stages of the runs, relaxing it in more advances stages of the search. The main goal of these experiments is determining if human-guidance results in a benefit, rather than the absolute quality of the results. Fig. 2–4 depict a comparison of the best solutions found by each of the approaches. As it can be seen, these best solutions are better in the case of the hgEA. Median values for the aEA are 22, 67 and 68 for each of the three instances, whereas the hgEA provide median values of 20.5, 64 and 64.5 (differences statistically significant at 0.05 level).

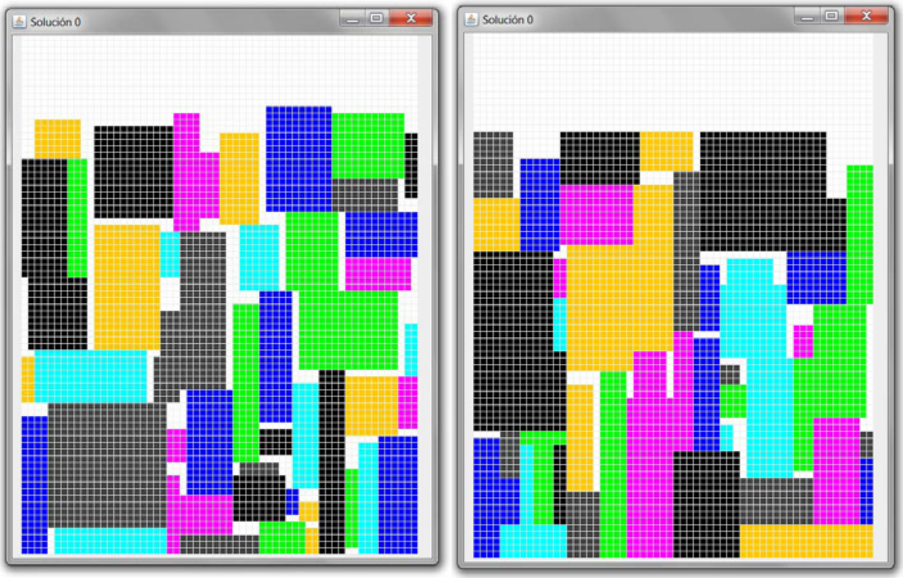


Fig. 4. Best solution found in the 49-object instance with the automated EA (left), and the human-guided EA (right)

5 Conclusions

This work has presented an initial progress report on the use of human-guided evolutionary algorithms for solving 2D packing problems. In this sense, the results are encouraging and hint the potential for user-centric techniques in this domain. A point of caution when interpreting the results is required though. Besides using a more ample test suite to analyze the scalability of this kind of techniques, it is required to analyze in a deeper way the changes the user interaction is exerting on the search dynamics and population diversity. It may be the case that these effects can be reverse-engineered and integrated into an automated EA approach. Then again, this would indicate the usefulness of the human guide, at least from a designer's point of view. Future work will try to confirm this.

Acknowledgements. This work is supported by project NEMESIS (TIN-2008-05941) of the Spanish Ministerio de Ciencia e Innovación, and project TIC-6083 of Junta de Andalucía.

References

1. Dyckhoff, H.: A typology of cutting and packing problems. European Journal of Operational Research 44, 145–159 (1990)

2. Hopper, E., Turton, B.: An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. European Journal of Operational Research 128, 34–57 (2001)
3. Cieliebak, M., Hall, A., Jacob, R., Nunkesser, M.: Sequential vector packing. Theoretical Computer Science 409(3), 351–363 (2008)
4. Hopper, E., Turton, B.: A review of the application of meta-heuristic algorithms to 2d regular and irregular strip packing problems. Artificial Intelligence Review 16, 257–300 (2001)
5. Hart, W.E., Belew, R.K.: Optimizing an arbitrary function is hard for the genetic algorithm. In: Belew, R.K., Booker, L.B. (eds.) Proceedings of the 4th International Conference on Genetic Algorithms, pp. 190–195. Morgan Kaufmann, San Mateo (1991)
6. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)
7. Davis, L.D.: Handbook of Genetic Algorithms. Van Nostrand Reinhold Computer Library, New York (1991)
8. Cotta, C., Fernández, A.J.: A hybrid GRASP – evolutionary algorithm approach to golomb ruler search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 481–490. Springer, Heidelberg (2004)
9. Banzhaf, W.: Interactive evolution. In: Back, T., Fogel, D., Michalewicz, Z. (eds.) Evolutionary Computation. Basic Algorithms and Operators, pp. 228–234. IoP, Bristol (2000)
10. Takagi, H.: Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. Proceedings of the IEEE (9), 1275–1296 (2001)
11. Parmee, I.C., Abraham, J.A.R., Machwe, A.: User-centric evolutionary computing: Melding human and machine capability to satisfy multiple criteria. In: Knowles, J., Corne, D., Deb, K., Chair, D.R. (eds.) Multiobjective Problem Solving from Nature. Natural Computing Series, pp. 263–283. Springer, Heidelberg (2008)
12. Hwang, I.: An efficient processor allocation algorithm using two-dimensional packing. Journal of Parallel and Distributed Computing 42(1), 75–81 (1997)
13. Burke, E., Kendall, G.: Comparison of meta-heuristic algorithms for clustering rectangles. Computers & Industrial Engineering 37(1-2), 383–386 (1999)
14. Falkenauer, E.: Genetic Algorithms and Grouping Problems. J. Wiley & Sons, Chichester (1998)
15. Chazelle, B.: The bottom left bin packing heuristic: an efficient implementation. IEEE Transactions on Computers 32, 697–707 (1983)
16. Burke, E., Kendall, G., Whitwell, G.: A new placement heuristic for the orthogonal stock-cutting problem. Operations Research 52, 697–707 (2004)
17. Alvarez-Valdes, R., Parreno, F., Tamarit, J.: Reactive grasp for the strip-packing problem. Computers & Operations Research 35(4), 1065–1083 (2008)
18. Feo, T., Resende, M.: Greedy randomized adaptive search procedures. Journal of Global Optimization 6, 109–133 (1995)
19. Prais, M., Ribeiro, C.: Reactive GRASP: an application to a matrix decomposition problem in TDMA traffic assignment. INFORMS Journal on Computing 12, 164–176 (2000)
20. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Berlin (1992)

Particle Swarm Optimisation for Open Shop Problems with Fuzzy Durations

Juan José Palacios[1], Inés González-Rodríguez[2],
Camino R. Vela[1], and Jorge Puente[1]

[1] A.I. Centre and Department of Computer Science,
University of Oviedo, Spain
{puente,crvela}@uniovi.es
http://www.di.uniovi.es/tc

[2] Department of Mathematics, Statistics and Computing,
University of Cantabria, Spain
ines.gonzalez@unican.es

Abstract. In this paper we confront a variation of the open shop problem where task durations are allowed to be uncertain and where uncertainty is modelled using fuzzy numbers. We propose a particle swarm optimization (PSO) approach to minimise the expected makespan using priorities to represent particle position, as well as a decoding algorithm to generate schedules in a subset of possibly active ones. Finally, the performance of the PSO is tested on several benchmark problems, modified so as to have fuzzy durations, compared with a memetic algorithm from the literature.

1 Introduction

The open shop scheduling problem, with an increasing presence in the literature, is a problem with clear applications in industry—consider for instance testing facilities, where units go through a series of diagnostic tests that need not be performed in a specified order and where different testing equipment is usually required for each test [19]. Given its NP-hardness for a number of machines $m \geq 3$ [19], practical approaches to solving it usually involve heuristic strategies: simulated annealing [1], tabu search [16], genetic algorithms with local search [7], ant colony optimization [2], etc.

To enhance the range of applications of scheduling, part of the research is devoted to model the uncertainty and vagueness pervading real-world situations. The approaches are diverse and, among these, fuzzy sets have been used in a wide variety of ways [4]. Incorporating uncertainty usually requires a significant reformulation of the problem and solving methods, in order that the problem can be precisely stated and solved efficiently and effectively. Some heuristic methods have so far been proposed for fuzzy flow and job shop problems, where uncertain durations are modelled via fuzzy intervals, e.g. [11], [15], [24], [21]. However, to the best of our knowledge, the open shop problem has received little attention in the fuzzy framework: in [14] fuzzy sets are used to represent flexible job start

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 362–371, 2011.

and due dates, in [18] a genetic algorithm is proposed to solve the open shop with fuzzy durations, and in [10] this genetic algorithm is combined with a local search method.

In the following, we consider the fuzzy open shop problem with expected makespan minimisation, denoted $FuzO||E[C_{max}]$ and propose a bio-inspired technique to solve it. The rest of the paper is organized as follows. In Section 2 we formulate the problem and introduce the notation used across the paper. Then, in Section 3, we describe the main components of the PSO algorithm. Section 4 reports results from the experimental study. Finally, in Section 5 we summarise the main conclusions and propose ideas for future work.

2 Open Shop Scheduling with Uncertain Durations

The *open shop scheduling problem*, or *OSP* in short, consists in scheduling a set of n jobs $J_1, \ldots, J_n$ to be processed on a set of m physical resources or machines $M_1, \ldots, M_m$. Each job consists of m tasks or operations, each requiring the exclusive use of a different machine for its whole processing time without preemption, i.e. all operations must be processed without interruption. In total, there are mn operations, $\{O_k, 1 \leq k \leq mn\}$. A solution to this problem is a *schedule* –an allocation of starting times for all operations– which is *feasible*, in the sense that all constraints hold, and is also optimal according to some criterion. Here, the objective will be minimising the makespan C_{max}, that is, the time lag from the start of the first operation until the end of the last one, a problem often denoted $O||C_{max}$ in the literature.

2.1 Uncertain Durations

In real-life applications, it is often the case that it is not known in advance the exact time it will take to process one operation and only some uncertain knowledge is available, for instance, an interval of possible durations, or a most likely duration with a certain error margin. Such knowledge can be modelled using a *triangular fuzzy number* or TFN, given by an interval $[n^1, n^3]$ of possible values and a modal value n^2 in it. For a TFN N, denoted $N = (n^1, n^2, n^3)$, the membership function takes the following triangular shape:

$$\mu_N(x) = \begin{cases} \frac{x-n^1}{n^2-n^1} & : n^1 \leq x \leq n^2 \\ \frac{x-n^3}{n^2-n^3} & : n^2 < x \leq n^3 \\ 0 & : x < n^1 \text{ or } n^3 < x \end{cases} \tag{1}$$

In the open shop, we essentially need two operations on processing times (fuzzy numbers), the sum and the maximum. These are obtained by extending the corresponding operations on real numbers using the *Extension Principle*. However, computing the resulting expression is cumbersome, if not intractable. For the sake of simplicity and tractability of numerical calculations, we follow [5] and approximate the results of these operations, evaluating the operation only on the three defining points of each TFN. It turns out that for any pair of TFNs M and N, the approximated sum $M + N \approx (m^1 + n^1, m^2 + n^2, m^3 + n^3)$ coincides with the actual sum of TFNs; this is not necessarily so for the maximum $\max\{M, N\} \approx (\max\{m^1, n^1\}, \max\{m^2, n^2\}, \max\{m^3, n^3\})$, although they have identical support and modal value.

The membership function of a fuzzy number can be interpreted as a possibility distribution on the real numbers. This allows to define its expected value [17], given for a TFN N by $E[N] = \frac{1}{4}(n^1 + 2n^2 + n^3)$. It coincides with the neutral scalar substitute of a fuzzy interval and the centre of gravity of its mean value [4]. It induces a total ordering $\leq_E$ in the set of fuzzy intervals [5], where for any two fuzzy intervals M, N $M \leq_E N$ if and only if $E[M] \leq E[N]$.

2.2 Fuzzy Open Shop Scheduling

If processing times of operations are allowed to be imprecise and such imprecision or uncertainty is modelled using TFNs, the resulting schedule is fuzzy in the sense that starting and completion times for each operation and hence the makespan are TFNs. Each TFN can be seen as a possibility distributions on the values that the time may take. Notice however that there is no uncertainty regarding the task processing ordering given by the schedule.

An important issue with fuzzy times is to decide on the meaning of "optimal makespan". It is not trivial to optimise a fuzzy makespan, since neither the maximum nor its approximation define a total ordering in the set of TFNs. Using ideas similar to stochastic scheduling, we follow the approach taken for the fuzzy job shop in [9] and use the total ordering provided by the expected value and consider that the objective is to minimise the expected makespan $E[C_{max}]$. The resulting problem may be denoted $FuzO||E[C_{max}]$.

3 Particle Swarm Optimization for the FOSP

Given the complexity of the open shop, different metaheuristic techniques have been proposed to solve the general m-machine problem. Among these, [12] describes two heuristic methods to obtain a list of operation priorities later used in a list-scheduling algorithm; [16] proposes a tabu search algorithm; [7] introduces a genetic algorithm hybridised with local search; a genetic algorithm using heuristic seeding is given in [20], and [23] proposes a solution based on particle swarm optimisation. As mentioned in Section 1, among these only a genetic algorithm [18] and its combination with local search [10] have been applied to the case where durations are fuzzy numbers.

PSO is a population-based stochastic optimisation technique inspired by bird flocking or fish schooling [13]. In PSO, each position in the search space corresponds to a solution of the problem and particles in the swarm cooperate to find the best position (hence best solution) in the space. Particle movement is mainly affected by three factors:

- Inertia: Velocity of the particle in the latest iteration.
- *pbest*: The best position found by the particle.
- *gbest*: The best position found by the swarm so far (the best *pbest*).

The potential solutions or particles fly through the problem space changing their position and velocity by following the current optimum particles *pbest* and *gbest*.

```
1. generate and evaluate the initial swarm.
2. compute gbest and pbest for each particle.
while no termination criterion is satisfied do
    for each particle $k$ do
        for each dimension $d$ do
            3. update velocity $v_d^k$.
            4. update position $x_d^k$.
        5. evaluate particle $k$.
        6. update pbest and gbest values.
return the schedule from the best particle evaluated so far;
```

Alg. 1: A generic PSO algorithm

Algorithm 1 describes the structure of a generic PSO algorithm. First, the initial swarm is generated and evaluated. Then the swarm evolves until a termination criterion is satisfied and in each iteration, a new swarm is built from the previous one by changing the position and velocity of each particle towards its *pbest* and *gbest* locations.

In [23], a PSO algorithm was proposed to solve the deterministic OSP which improved the best published results. Here we shall extend this algorithm to the fuzzy framework and test it compared to the state-of-the-art methods.

3.1 Position Representation and Evaluation

Following [23], we use a priority-based representation for particle positions. Thus a schedule is encoded as a priority matrix $X^k = (x_{ij}^k)_{i=1\dots m, j=1\dots n}$, where x_{ij}^k denotes the priority of operation o_{ij}, the task of job j processed on machine i. An operation with smaller x_{ij}^k has a higher priority to be scheduled.

If we represent an OSP solution as a task processing order S, which is a permutation of tasks, we can transfer this permutation to a priority matrix and viceversa. For instance, given the following solution:

$$S = \left(o_{11}\; o_{13}\; o_{23}\; o_{12}\; o_{31}\; o_{33}\; o_{21}\; o_{23}\; o_{22}\right)$$

a particle in the space can be obtained by randomly setting x_{ij} in the interval $(p - 0.5, p + 0.5)$ where p is the location of o_{ij} in S. Therefore, the operation with smaller x_{ij} has higher priority to be scheduled. The above permutation list can be transferred to:

$$X^k = \begin{pmatrix} 1.2 & 4.0 & 1.7 \\ 6.6 & 9.4 & 2.7 \\ 5.3 & 7.9 & 6.4 \end{pmatrix}$$

```
while $\Omega \neq \emptyset$ do
    $f^* \leftarrow \min_{o_{ij} \in \Omega}\{E[f_{ij}]\}$.
    $s^* \leftarrow \min_{o_{ij} \in \Omega}\{E[s_{ij}]\}$.
    Identify the conflict set $O \leftarrow \{o_{ij} : E[s_{ij}] < s^* + \delta \times (f^* - s^*), o_{ij} \in \Omega\}$.
    Choose the operation $o^*_{ij}$ from $O$ with smallest $x^k_{ij}$.
    Schedule the operation $o^*$.
    $\Omega \leftarrow \Omega - \{o^*\}$.
```

Alg. 2: The *pFG&T* algorithm

Decodification of a particle may be done in different ways. For the crisp job shop and by extension for the open shop, it is common to use the G&T algorithm [6], which is an active schedule builder. A schedule is *active* if one task must be delayed for any other one to start earlier. Active schedules are good in average and, most importantly, the space of active schedules contains at least an optimal one, that is, the set of active schedules is *dominant*. For these reasons it is worth to restrict the search to this space. In [7] a narrowing mechanism was incorporated to the G&T algorithm in order to limit machine idle times by means of a delay parameter $\delta \in [0, 1]$, thus searching over the space of so-called parameterised active schedules. In the deterministic case, for $\delta < 1$ the search space is reduced so it may no longer contain optimal schedules and, at the extreme $\delta = 0$ the search is constrained to non-delay schedules, where a resource is never idle if a requiring operation is available. This variant of G&T has been applied in [23] to the deterministic OSP, under the name "parameterized active schedule generation algorithm".

In Algorithm 2 we propose an extension of parameterised G&T to the case of fuzzy processing times, denoted *pFG&T*. It should be noted that, due to the uncertainty in task durations, even for $\delta = 1$, we cannot guarantee that the produced schedule will indeed be active when it is actually performed (and tasks have exact durations). We may only say that the obtained fuzzy schedule is *possibly active*. Throughout the algorithm, Ω detones the set of the operations that have not been scheduled, X^k the priority matrix, s_{ij} the starting time of the operation o_{ij} and f_{ij} the completion time of the operation o_{ij}.

Notice that the *pFG&T* algorithm may change the task processing order given by the particle position. Therefore the PSO does not record in *gbest* and *pbest* the best positions found so far, but rather the best operation sequences of the schedules generated by the decoding operator.

3.2 Particle Movement and Velocity

Particle velocity is traditionally updated depending on the distance to *gbest* and *pbest*. Instead, this PSO only considers whether the position value x^k_{ij} is larger or smaller than $pbest^k_{ij}$ ($gbest_{ij}$). For any particle, its velocity is represented by an array of the same length as the position array where all the values are in the set $\{-1, 0, 1\}$. Updating is controlled by the inertia weight w at the beginning of each iteration as follows. For each particle k and operation o_{ij} (in the following,

```
for each dimension $d$ do
    generate random value $rand \sim U(0,1)$.
    if $v_d^k \neq 0$ and $rand \geq w$ then
        $v_d^k \leftarrow 0$.
    if $v_d^k = 0$ then
        generate random value $rand \sim U(0,1)$.
        if $rand \leq C_1$ then
            if $pbest_d^k \geq x_d^k$ then
                $v_d^k \leftarrow 1$.
            else
                $v_d^k \leftarrow -1$.
            generate random value $rand_2 \sim U(0,1)$.
            $x_d^k \leftarrow pbest_d^k + rand_2 - 0.5$.
        if $C_1 < rand \leq C_1 + C_2$ then
            if $gbest_d \geq x_d^k$ then
                $v_d^k \leftarrow 1$.
            else
                $v_d^k \leftarrow -1$.
            generate random value $rand_2 \sim U(0,1)$.
            $x_d^k \leftarrow gbest_d + rand_2 - 0.5$.
    else
        $x_d^k \leftarrow x_d^k + v_d^k$.
```

Alg. 3: Particle movement

to simplify, dimension d), if $v_d^k \neq 0$, v_d^k will be set to 0 with probability $1 - w$, meaning that if x_d^k was either increasing or decreasing, x_d^k stops at this iteration with probability $1 - w$. Otherwise, if $v_d^k = 0$, with probability C_1, v_d^k and x_d^k will be updated depending on $pbest_d^k$ and with probability C_2 they will be updated depending on $gbest_d$, always introducing an element of randomness and where C_1 and C_2 are constants between 0 and 1 such that $C_1 + C_2 \leq 1$. The details on how the updating of particle k takes place are given in Algorithm 3.

Position mutation. After a particle moves to a new position, we randomly choose an operation and then mutate its priority value x_d^k disregarding v_d^k. As in [23], for a problem of size $n \times m$, if $x_d^k < (nm/2)$, x_d^k will take a random value in $[mn - n, mn]$, and $v_d^k = 1$. Otherwise, if $x_d^k > (nm/2)$, x_d^k will take a random value in $[0, n]$ and $v_d^k = -1$.

Diversification strategy. If all particles have the same *pbest* solutions, they will be trapped into local optima. To prevent such situation, a diversification strategy is proposed in [23] that keeps the *pbest* solutions different. In this strategy, the *pbest* solution of each particle is not the best solution found by the particle itself, but one of the best N solutions found by the swarm so far, where N is the size of the swarm. Once any particle generates a new solution, the *pbest* solutions will be updated in these situations:

– if the makespan of the particle solution is equal to any *pbest* solution, replace that *pbest* solution with the new particle solution;
– if the makespan of the particle solution is less than the worst *pbest* solution and different from all *pbest* solutions, set the worst *pbest* solution equal to the particle solution.

4 Experimental Results

For the experimental study we use the test bed given in [10], where the authors follow [5] and generate a set of fuzzy problem instances from well-known benchmark problems from [3]. Given a crisp problem instance, each crisp processing time t is transformed into a symmetric fuzzy processing time $p(t)$ such that its modal value is $p^2 = t$ and p^1, p^3 are random values, symmetric w.r.t. p^2 and generated so the TFN's maximum range of fuzziness is 30% of p^2. By doing this, the optimal solution to the crisp problem provides a lower bound for the expected makespan of the fuzzified version [5]. The original problem instances consist of 6 families, denoted J3, J4,. . . , J8, of sizes 3×3, 4×4,. . . ,8×8, containing 8 or 9 instances each. From each crisp problem instance 10 fuzzy versions were generated, so in total there are 520 problem instances. The obtained benchmarks for the fuzzy open shop are available at `http://www.di.uniovi.es/tc`.

In [10], the authors propose a neighbourhood structure which combined with the GA from [18] provides a MA that not only obtains better solutions but is also more "reliable" in the sense that there is less variability in quality solution across different executions. In this experimental study we compare our PSO with this MA.

For the PSO we take the best values for each parameter obtained after a parameter analysis in [23]: swarm size $N = 60$, $C_1 = 0.7$, $C_2 = 0.1$, and inertia weight w linearly decreasing from 0.9 to 0.3. The number of iterations has been adapted for each problem size so as to obtain similar running times to the MA. Regarding the filtering mechanism of the search space given in the schedule generator, the efficiency of this reduction depends of the problem size [8], in fact our experimentation suggest taking $\delta = 1$ (no reduction) in small instances (families J3 and J4) and $\delta = 0.25$ in larger ones.

To evaluate its performance, we run the proposed PSO 30 times for each problem instance, recording the best, average expected makespan values across these 30 runs. Table 1 shows a summary of the results, with average values across 30 executions on each the 80–90 instances of the same size (detailed results for each problem would require 520 rows). It contains three columns for each method, PSO and MA, showing the Average of the Best values (AoB), the Average of the Average values (AoA), and the Average CPU Times in seconds ($Time$). We can see that both the PSO and the MA perform equally well on the small (3×3, 4×4) problems (the AoA value of PSO is slightly worse in the J4 family). As the problem size increases, also does increase the difference in solution quality between the PSO and the MA, with the former obtaining better results.

Table 1. Comparison between PSO and MA

Problem Family	PSO			MA		
	$E[C_{max}]$		$Time$	$E[C_{max}]$		$Time$
	AoB	AoA		AoB	AoA	
J3	1063.1	1063.1	0.06	1063.1	1063.1	0.13
J4	1048.8	1057.2	0.11	1048.8	1050.4	0.23
J5	1028.7	1029.6	0.70	1030.8	1044.9	1.29
J6	1033.3	1036.0	4.86	1033.9	1052.1	7.57
J7	1036.0	1041.6	14.84	1044.1	1067.5	14.46
J8	1031.6	1039.0	30.69	1045.5	1068.2	26.15

Table 2. Average relative makespan error (in %) for sets of problems of size 8 × 8

Problem	PSO		MA	
	B	A	B	A
`j8-per0-1`	6.415	7.586	9.395	12.406
`j8-per0-2`	7.292	7.858	8.806	11.646
`j8-per10-0`	4.467	5.189	5.793	8.308
`j8-per10-1`	2.405	3.293	4.03	6.586
`j8-per10-2`	2.380	3.602	4.603	7.077
`j8-per20-0`	1.255	1.844	1.870	3.672
`j8-per20-1`	0.038	0.204	0.185	1.379
`j8-per20-2`	1.062	1.621	1.733	3.466

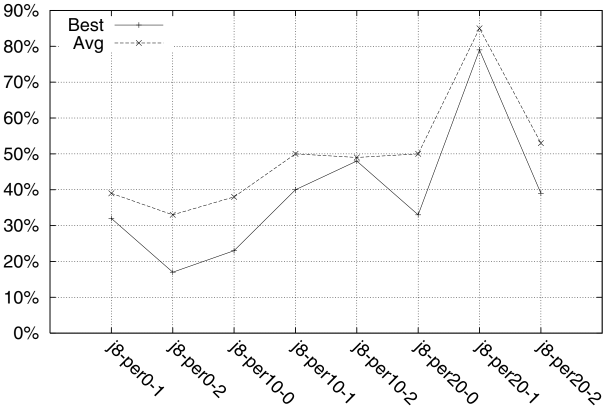


Fig. 1. Percentage of reduction in average relative error of PSO w.r.t. MA

More detailed results are presented in Table 2, where each row corresponds to one set of ten fuzzy versions of one of the crisp instances of size 8 × 8. It shows relative makespan errors w.r.t. a lower bound for the expected makespan, which is 1000 for all problem instances. As expected, the PSO compares favourably with MA in all instances. Notice as well that the relative errors for the best (B) and average (A) solution do not differ greatly, suggesting that the PSO is quite stable. Figure 1 illustrates the reduction of the makespan error in average for each set of fuzzy problems; we can observe this is higher for mean values, which are more significant in stochastic algorithms.

5 Conclusions and Future Work

We have considered an open shop problem with uncertain durations modelled as triangular fuzzy numbers, $FuzO||E[C_{max}]$, and have proposed a particle swarm optimization technique to solve this problem. The PSO has obtained good results both in terms of relative makespan error and also in comparison to a memetic algorithm from the literature. These promising results suggest directions for future work. First, the PSO should be tested on more difficult problems, fuzzy versions of other benchmark problems from the literature. Also, the PSO provides a solid basis for the development of more powerful hybrid methods, in combination with local search techniques, an already successful approach in fuzzy job shop problems [22].

Acknowledgments

This work is supported by the Spanish Ministry of Science and Education under research grant MEC-FEDER TIN2010-20976-C02-02.

References

1. Andresen, M., Bräsel, H., Mörig, M., Tusch, J., Werner, F., Willenius, P.: Simulated annealing and genetic algorithms for minimizing mean flow time in an open shop. Mathematical and Computer Modelling 48, 1279–1293 (2008)
2. Blum, C.: Beam-ACO—hybridizing ant colony optimization with beam search: an application to open shop scheduling. Computers & Operations Research 32(6), 1565–1591 (2005)
3. Brucker, P., Hunrink, J., Jurisch, B., Wöstmann, B.: A branch & bound algorithm for the open-shop problem. Discrete Applied Mathematics 76, 43–59 (1997)
4. Dubois, D., Fargier, H., Fortemps, P.: Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. European Journal of Operational Research 147, 231–252 (2003)
5. Fortemps, P.: Jobshop scheduling with imprecise durations: a fuzzy approach. IEEE Transactions of Fuzzy Systems 7, 557–569 (1997)
6. Giffler, B., Thompson, G.L.: Algorithms for solving production scheduling problems. Operations Research 8, 487–503 (1960)

7. Gonçalves, J., Mendes, J., de M, R.M.: A hybrid genetic algorithm for the job shop scheduling problem. European Journal of Operational Research 167, 77–95 (2005)
8. González, M.A., Sierra, M., Vela, C.R., Varela, R.: Genetic algorithms hybridized with greedy algorithms and local search over the spaces of active and semi-active schedules. In: Marín, R., Onaindía, E., Bugarín, A., Santos, J. (eds.) CAEPIA 2005. LNCS (LNAI), vol. 4177, pp. 231–240. Springer, Heidelberg (2006)
9. González-Rodríguez, I., Puente, J., Vela, C.R.: Sensitivity analysis for the job shop problem with uncertain durations and flexible due dates. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2007. LNCS, vol. 4527, pp. 538–547. Springer, Heidelberg (2007)
10. González-Rodríguez, I., Palacios, J.J., Vela, C.R., Puente, J.: Heuristic local search for fuzzy open shop scheduling. In: Proceedings IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2010, pp. 1858–1865. IEEE, Los Alamitos (2010)
11. González Rodríguez, I., Puente, J., Vela, C.R., Varela, R.: Semantics of schedules for the fuzzy job shop problem. IEEE Transactions on Systems, Man and Cybernetics, Part A 38(3), 655–666 (2008)
12. Guéret, C., Prins, C.: Classical and new heuristics for the open-shop problem: A computational evaluation. European Journal of Operational Research 107, 306–314 (1998)
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Press, New Jersey (1995)
14. Konno, T., Ishii, H.: An open shop scheduling problem with fuzzy allowable time and fuzzy resource constraint. Fuzzy Sets and Systems 109, 141–147 (2000)
15. Lei, D.: Pareto archive particle swarm optimization for multi-objective fuzzy job shop scheduling problems. International Journal of Advanced Manufacturing Technology 37, 157–165 (2008)
16. Liaw, C.F.: A tabu search algorithm for the open shop scheduling problem. Computers and Operations Research 26, 109–126 (1999)
17. Liu, B., Liu, Y.K.: Expected value of fuzzy variable and fuzzy expected value models. IEEE Transactions on Fuzzy Systems 10, 445–450 (2002)
18. Palacios, J.J., Puente, J., Vela, C.R., González-Rodríguez, I.: A genetic algorithm for the open shop problem with uncertain durations. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5601, pp. 255–264. Springer, Heidelberg (2009)
19. Pinedo, M.L.: Scheduling. Theory, Algorithms, and Systems, 3rd edn. Springer, Heidelberg (2008)
20. Puente, J., Diez, H., Varela, R., Vela, C., Hidalgo, L.: Heuristic Rules and Genetic Algorithms for Open Shop Scheduling Problem. In: Conejo, R., Urretavizcaya, M., Pérez-de-la-Cruz, J.-L. (eds.) CAEPIA/TTIA 2003. LNCS (LNAI), vol. 3040, pp. 394–403. Springer, Heidelberg (2004)
21. Puente, J., Vela, C.R., González-Rodríguez, I.: Fast local search for fuzzy job shop scheduling. In: Proceedings of ECAI 2010, pp. 739–744. IOS Press, Amsterdam (2010)
22. Sha, D.Y., Cheng-Yu, H.: A modified parameterized active schedule generation algorithm for the job shop scheduling problem. In: Proceedings of the 36th International Conference on Computers and Industrial Engineering, ICCIE 2006, pp. 702–712 (2006)
23. Sha, D.Y., Cheng-Yu, H.: A new particle swarm optimization for the open shop scheduling problem. Computers & Operations Research 35, 3243–3261 (2008)
24. Tavakkoli-Moghaddam, R., Safei, N., Kah, M.: Accessing feasible space in a generalized job shop scheduling problem with the fuzzy processing times: a fuzzy-neural approach. Journal of the Operational Research Society 59, 431–442 (2008)

Design of Emergent and Adaptive Virtual Players in a War RTS Game

José A. García Gutiérrez, Carlos Cotta, and Antonio J. Fernández Leiva

Dept. Lenguajes y Ciencias de la Computación, ETSI Informática,
Campus de Teatinos, Universidad de Málaga,
29071 Málaga – Spain
{ccottap,afdez}@lcc.uma.es

Abstract. Basically, in (one-player) war Real Time Strategy (wRTS) games a human player controls, in real time, an army consisting of a number of soldiers and her aim is to destroy the opponent's assets where the opponent is a virtual (i.e., non-human player controlled) player that usually consists of a pre-programmed decision-making script. These scripts have usually associated some well-known problems (e.g., predictability, non-rationality, repetitive behaviors, and sensation of artificial stupidity among others). This paper describes a method for the automatic generation of virtual players that adapt to the player skills; this is done by building initially a model of the player behavior in real time during the game, and further evolving the virtual player via this model in-between two games. The paper also shows preliminary results obtained on a one-player wRTS game constructed specifically for experimentation.

1 Introduction and Related Work

In an era in which computing power has boosted the graphical quality of videogames, players have turned their attention to other aspects of the game. In particular, they mostly request opponents exhibiting intelligent behavior. However, intelligence does not generally equate to playing proficiency but rather to interesting behaviors [1]. This issue is specifically relevant in real-time strategy (RTS) games that often employ two kinds of artificial intelligence (AI) [2]: one represented by a *virtual player* (VP, or non-player character – NPC) making decisions on a set of units (i.e., warriors, machines, etc.), and another one corresponding to the small units (usually with little or no intelligence). The design of these AIs is a complex task, and the reality in the industry is that in most of the RTS games, the NPC is basically controlled by a fixed script that has been previously programmed based on the experience of the designer/programmer. This script often comprises hundreds of rules, in the form IF THE GAME IS IN STATE S THEN UNIT U SHOULD EXECUTE ACTION A, to control the behavior of the components (i.e., units) under its control. This arises known problems: for instance, for players, the opponent behavior guided by fixed scripts can become predictable for the experienced player. Also, for AI programmers, the design of virtual players (VPs) can be frustrating because the games are becoming more

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 372–382, 2011.

and more complex with hundreds of different situations/states and therefore it is not easy to predict all the possible situations that could potentially happen and even more difficult to decide which is the most appropriate actions to take in such situations. As consequence, many RTS games contain 'holes' in the sense that the game stagnates or behaves incorrectly under very specific conditions (these problems rely on the category of 'artificial stupidity' [1]). Thus the reality of the simulation is drastically reduced and so too the interest of the player;

In addition, there are very interesting research problems in developing AI for Real-Time Strategy (RTS) games including planning in an uncertain world with incomplete information, learning, opponent modeling, and spatial and temporal reasoning [3]. This AI design is usually very hard due to the complexity of the search space (states describe large playing scenarios with hundreds of units simultaneously acting). Particular problems are caused by the large search spaces (environments consisting of many thousands of possible positions for each of hundreds, possibly thousands, of units) and the parallel nature of the problem - unlike traditional games, any number of moves may be made simultaneously [4]. Qualitative spatial reasoning (QSR) techniques can be used to reduce complex spatial states (e.g., using abstract representations of the space [5]). Regarding evolutionary techniques, a number of biologically-inspired algorithms and multi-agent based methods have already been applied to handle many of the mentioned problems in the implementation of RTS games [6,7,8,9,10,11,12,13].

Even in the case of designing a very good script for VPs, the designer has to confront another well known problem: the VP behavior is usually fixed and rarely adapts to the level (i.e., skills) of the player. In fact the player can lose interest in the game because either she is able to beat all the opponents in each level, or the virtual player always beat her. In this sense, the design of interesting (e.g., non-predictable) NPCs is not the only challenge though. It also has to adapt to the human player, since she might lose interest in the game otherwise (i.e., if the NPCs are too easy or too hard to beat).

There are many benefits attempting to build adaptive learning AI systems which may exist at multiple levels of the game hierarchy, and which evolve over time. This paper precisely deals with the issue of generating behaviors for the virtual player that evolve in accordance with the player's increasing abilities. This behavior emerges according to the player skill, and this emergent feature can make a RTS game more entertaining and less predictable in the sense that emergent behavior is not explicitly programmed but simply happens [14]. The attainment of adjustable and emergent virtual players consists here of a process of two stages that are iteratively executed in sequence: (1) a behavior model of the human player is created in real time during the execution of a game, and further the virtual player is evolved off-line (i.e., in between two games) via evolutionary algorithms till a state that it can compete with the player (not necessarily to beat her but to keep her interest in the game). This approach has been applied on a RTS game constructed specifically for experimentation, and we report here our experience.

2 The Game

Here we describe the basic elements that compose our wRTS game.

Scenario. It will also be called indistinctly *region* or *world* and consists of a two-dimensional non-toroidal heterogeneous hostile and dynamic grid-world. The world is *heterogeneous* because the terrain is not uniform, *hostile* because there exist a virtual army whose mission is to destroy the human player-controlled army, and *dynamic* because the game conditions change depending on the actions taken by units of both armies. Regarding the heterogeneity of the terrain, each grid in the region can have one of the three following values: passable (each uni can traverse this grid), impassable (no unit can traverse it), and semi-impassable (there is a penalization of 1 point of energy - see below -to traverse it).

Army. There are two armies (also called indistinctly teams, represented by spiders and ladybirds) with a number of units (also called indistinctly soldiers or agents) and a flag to be defended. One army is controlled by the human player whereas the other one is guided by some kind of artificial intelligence (AI).

Given a unit u placed in a position (x_u, y_u) in the grid, its *visual range* (VR_u) embraces any position (x, y) that is placed to a maximum distance ϕ, that is to say, $\mathrm{VR}_u = \{(x, y) \mid \sqrt{(x_u - x)^2 + (y_u - y)^2} \leq \phi\}$. Initially, each agent only knows the grids that belongs to its visual range. In fact, during the game the scenario is not completely known for an army and only those regions that were already visualized by its agents are known. The global information that each army (independently if this is controlled by the AI or the player) has is the sum of all the information of their constituent soldiers. Note that (human or virtual) players only know the rival flag position if this has been detected in a grid by some of its agents previously. Also, each unit interacts with the environment by executing a number of actions (see below) and has certain level of health and energy that decreases with these interactions. The initial values for the health and energy of each soldier are 100 and 1000 respectively.

Fights. Full body combat can be executed at the soldier level by assigning a random value (in the interval [0,1]) to each unit involved in a fight with a rival soldier; the energy level of the unit with lowest value decreases 1 unit. A unit dies when its energy is zero. All the combats between soldiers are executed in sequence in one turn of the game.

Decision-making. Along the game, the role of players is to make decisions with respect to the actions that the units of their respective associated armies should take. More specifically, both the virtual and human player can select a set of agents and further assign it a specific order that basically consists of a common action that each unit in this set has to execute.

Actions. The game is executed in turns; every turn, each soldier executes the last order received by its player. Six actions are possible in this game:

- Move forward enemy: if applied to unit u, then this has to move towards the closer rival soldier according to VR_u, otherwise (i.e., if no rival soldier exists

in VR_u) agent u has to move in direction to the region where there are more enemies according to the global information of the army.
– Group/Run away: the soldier has to group with other team mates placed inside its visual range; if there are not mates in this range, then the agent has to move towards the position where its flag is placed.
– Move forward objective: the agent advances towards the opponent's flag position if this is known, otherwise it moves randomly.
– No operation: execute no operation.
– Explore: the soldier moves towards a non-explored-yet region of the map.
– Protect flag: The soldier has to move towards the position of its own flag; if it is already near to it, unit just should scout the zone.

Perceptions. Each unit knows its specific situation in the game from the perceptions that receives in its visual range. In our game, the *agent state* is determined by its health (measured in three range values: low, medium and high), and the following boolean perceptions:

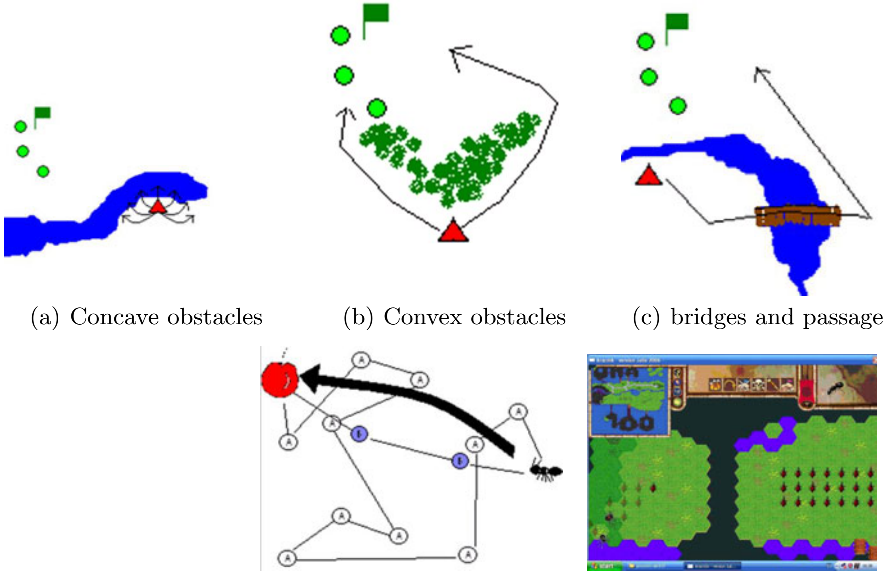
1. Advantage state? (S_a): "the unit is into a handicap situation". This perception is true if, according to the information in its visual range, the number of mate soldiers is higher than the number of rival units, and false otherwise.
2. Under attack? (U_a): this perception is true of the soldier is involved in a full body combat with another opponent unit, and false otherwise.
3. Objective visible? (O_v): the opponent's flag is visible in its visual range.

Game objective. The army that first captures the rival flag is the winner; if this situation is never reached after a number of turns, then the winner is the player whose army inflicts a higher damage to its rival army and where the damage is measured as number of dead units.

2.1 Notes on Specific Issues

Observe that 'fighting' is not a specific action to be executed by the army units. This should not be surprising because 'fighting' is not usually controlled by players in standard RTS games as this action should often is executed at the unit level and not at the army level; in our game this action is automatically executed when a unit and a rival soldier meet in any grid of the scenario.

Also, note that five of the six possible actions that units can execute require to make movements in the game world. This basically means that some kind of pathfinding should be processed to do the movement realistic. Classical algorithms such as A* were not good candidates for this task since, as already indicated, most of the scenario is not known for the army units. A practical solution was to let units make a good-enough movement depending on its scenario knowledge; several cases were covered: *Concave obstacles*: the unit moves by considering different angles with respect to its position in the scenario; *Convex obstacles*: the unit moves by following the obstacle contour in a specific direction; and *Bridges and narrow passages*: a simple ant-colony algorithm was executed.



(a) Concave obstacles (b) Convex obstacles (c) bridges and passages

(d) Ant-colony (e) Game

Fig. 1. (a)-(c) Movement examples; (d) illustration of classical execution of ant-colony algorithm used for pathfinding; (e) game screenshot (black cells represent non-explored regions yet, blue cells impassable grids, and green cells passable regions in the scenario

Due to space limitations we will not explain them in details as this is beyond the objective of this paper. In any case, Figures 1(a),(b) and (c) show illustrative examples of the cases, Figure 1(d) illustrates the classical behavior of our ant colony algorithms, and Figure 1(e) displays a screenshot of our wRTS game[1] titled *aracnia*. Note also that a description of the technical issues of the game (e.g., implementation of basic actions, graphical considerations, input/output interface, scene rendering and memory management, among others) is beyond the scope of this paper. In the following we will focus on the process of designing virtual players.

3 Virtual Player Design

Here we describe our proposal to design adaptive virtual players in our wRTS.

3.1 General Issues

Our aim is to generate controllers governing automatically (i.e., without human intervention) the behavior of an entire team (i.e., a set of autonomous

[1] A prototype version of this game can be unloaded from `http://www.lcc.uma.es/∼afdez/aracnia`.

agents/soldiers). An army controller can be viewed as a set of rules that will control the reactions of its agents according to the current state of play. Depending on its situation, the agent executes a particular action that may modify the state of the agent itself. The global team strategy emerges as the result of the sum of the specific behavior of all the agents.

However, the definition of specific strategies for each agent is complex, requires a profound knowledge of the required behavior of each agent, and results in predictable behavior. Moreover, devising independent controllers for each unit in a RTS game is costly; in fact this is not realistic as requires a high computation effort that surely decreases the quality of the real-time rendering of the game. One solution consists of designing global team strategies; however, again this is very complex as the programmer has to cope with too many possibilities that arise from the different interactions among all the agents of both armies. We have opted by a more realistic process: all the soldiers will be managed by the same controller, and in any instant of the game, the team behavior will be the result of summing all the action taken by each of its constituent units. This means that we have to produce just one controller to devise a global team strategy. Note however that this does not mean all the agents execute the same action because the action to be executed by a soldier will depend on its particular situation in the match.

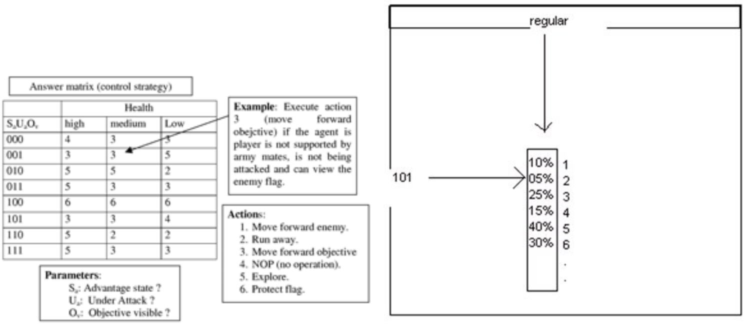
3.2 A Two Phases Process

The procedure described in the following has the aim of creating a virtual player whose behavior auto-evolves according to the skills that human player exhibits along the game. Algorithm 1 displays the general schema of the process: initially (in the first game), the virtual player (VP) is constructed as an expert system (i.e., a rule-based prototype (RBP)) that comprises tens of rules trying to cover all the possible situations in which an agent might be involved. This system was the result of a number of trials designed from the experiences of the authors playing wRTS games. Then, let $\mathbb{N}_h$ be $\{1, \ldots, h\}$ henceforth, and assume the human player will play a number of $\wp$ games, the procedure consists of two phases that are sequentially executed. Firstly, a *player modeling phase* is conducted; this step is described in Section 3.3 and basically consists of building a model of the behavior that human player exhibits during the game (i.e., on-line). The second phase, described in Section 3.4, is devoted to construct the virtual player by evolving a strategy controlling the functioning of a virtual army unit via an evolutionary algorithm (EA). In the following we describe in details both phases. Firstly, we discuss how the virtual player is internally encoded as this is important to understand the functioning of the whole schema.

Representation: the virtual player contains a NPC army strategy, and its internal encoding corresponds with both the rule-based expert system and any individual in the EA population. In the following we will refer to 'individual' in a general sense. Every individual represents a team strategy, that is to say, a common set of actions that each army unit has to execute under its specific

Algorithm 1. PMEA (VP)

```
VP ← RBP;   // Rule-based expert system
for $i \in \mathbb{N}_\wp$ do
    PlayerModel ← PlayGameOn(VP);   // Player modeling (PM)
    VP ← EA(PlayerModel,VP);   // Evolutionary optimization (EA)
end for
return VP;
```



Answer matrix (control strategy)

	Health		
$S_a U_a O_v$	high	medium	Low
000	4	3	3
001	3	3	5
010	5	5	2
011	5	3	3
100	6	6	6
101	3	3	4
110	5	2	2
111	5	3	3

Parameters:
S_a: Advantage state ?
U_a: Under Attack ?
O_v: Objective visible ?

Example: Execute action 3 (move forward obejctive) if the agent is player is not supported by army mates, is not being attacked and can view the enemy flag.

Actions:
1. Move forward enemy.
2. Run away.
3. Move forward objective
4. NOP (no operation).
5. Explore.
6. Protect flag.

Fig. 2. Example of an arbitrary encoding (left), and extended answer matrix (right)

situation in the environment. An individual is represented as a vector v of k cells where k is the number of different situations in which the agent can be in the game, and $v[i]$ (for $0 \leq i < k$) contains the action to be taken in situation i. In other words, assuming m perceptions, where the perception p_j (for $0 \leq j \leq m-1$) can have k_j possible values, then $k = k_0 * k_1 * \ldots * k_{m-1}$ and the cell:

$$v[e_{m-1}+e_{m-2}*k_{m-1}+e_{m-3}*(k_{m-1}*k_{m-2})+e_{m-4}*(k_{m-1}*k_{m-2}*k_{m-3})+\ldots\ldots]$$

contains the action to be executed by a specific unit when its perceptions $p_0, p_1, \ldots, p_{m-1}$ have the values $e_0, e_1, \ldots, e_{m-1}$ respectively. As indicated in Section 2, the state of a unit in our wRTS game is determined by three boolean perceptions and its energy level (with 3 different values) so that the encoding of an individual consists of a vector with 24 genes; this vector will be called *answer matrix*, and each cell in the vector will contain an action. Remember that 6 possible actions can be executed in our wRTS and thus the search space (i.e., the number of different strategies that can be generated from this representation) is 6^{24}. Figure 2(left) displays an example of a possible encoding. The optimal solution (if it exists) would be that strategy which always select the best action to be executed for the agents under all possible environmental conditions. In fact, this vast search space makes this problem impracticable for exact methods and justifies the use of evolutionary algorithms.

3.3 On-Line Phase: User Modeling

The player behavior will be modeled as a ruled-based strategy encoded as explained above. This process requires collecting, during the execution of the game

Algorithm 2. EA(PModel, VP)) // PModel = Player Model, VP = Virtual Player

```
1  for $i \in \mathbb{N}_{popsize-1}$ do
2      $pop_i$ ←RANDOM-SOLUTION();
3  end for
4  $pop_{\mathrm{popsize}}$ ←VP;
5  $i \leftarrow 0$;
6  while $i < MaxGenerations$ do
7      RANK-POPULATION ($pop$); // sort population according to fitness
8      $parent_1$ ←SELECT ($pop$); // roulette wheel
9      $parent_2$ ← SELECT ($pop$);
10     if Rand$[0, 1] < p_X$ then // recombination is done
11         $(child_1, child_2)$ ← RECOMBINE $(parent_1, parent_2)$;
12     else
13         $(child_1, child_2) \leftarrow (parent_1, parent_2)$;
14     end if
15     $child_1$ ← MUTATE $(child_1, p_M)$; // $p_M$ = mutation probability
16     $child_2$ ← MUTATE $(child_2, p_M)$;
17     $fitness_1$ ← PLAYGAMEOFF$(PModel, child_1)$;
18     $fitness_2$ ← PLAYGAMEOFF$(PModel, child_2)$;
19     $pop$ ← REPLACE$(pop, child_1, child_2)$; // (popsize + 2) replacement
20     $i \leftarrow i + 1$;
21 end while
22 return best solution in $pop$;
```

(i.e., in real time) the actions that human player executes, recording additionally the specific conditions under which these are performed. At the end of this process we generate an *extended answer matrix* (v^+) that is an answer matrix (i.e., an individual encoding v) where each cell $v^+[i]$ (for $0 \le i < k$) represents now a vector of 6 positions (one per action) and $v^+[i][a]$ (for some action $a \in [1, 6]$) contains the probability that human player executes action a under the environment conditions (i.e., the state) associated to cell $v[i]$. Figure 2(right) displays an example that shows the probability of executing each of the 6 possible actions in a specific situation of the unit (i.e., the soldier has medium energy, is in an advantage state, is not suffering an attack, and knows where the opponent flag is placed). This extended answer matrix is finally used to design the virtual player as follows: VP$[i] = argmax_{a\in[1,6]}\{v^+[i][a]\}$, for all possible situations i.

3.4 Off-Line Phase: Evolutionary Optimization

Algorithm 2 shows the basic schema of our EA. The initial population is randomly generated except one individual that is charged with the virtual player (lines 1-4). Then a classical process of evolution (lines 6-21) is performed and the best solution in the population is finally returned (draws are broken randomly).

Evaluating the fitness of an individual x requires the simulation (off-line) of the game between the player model and the virtual player strategy encoded in x. The fitness function depends on the statistical data collected at the end of the simulation. The higher the number of statistical data to be collected, the higher the computational cost will be. A priori, a good policy is to consider a limited number of data. Four data were used in our experiments during off-line evaluation: A: Number of deaths in the human player army; B: number of deaths in virtual player army; C: number of movements; and D: victory degree (i.e., 1 if virtual player wins and 2 otherwise). Fitness function was then defined as $fitness(x) = \frac{10000*(A-B)}{C*D}$; higher the fitness value, better the strategy. This fitness was coded to evolve towards aggressive solutions.

4 Experimental Analysis

The experiments were performed using two algorithms: our initial expert system (RBP), and the algorithm PMEA (i.e., player modeling + EA) presented in Algorithm 1. As to the PMEA, the EA uses $popsize = 50$, $p_X = .7$, $p_M = .01$, and MaxGenerations = 125; mutation is executed as usual at the gene level by changing an action to any other action randomly chosen. Three different scenarios where created for experimentation: (1) A map with size 50 × 50 grids, 48 agents in VP army, and 32 soldiers in the human player (HP) team; (2) a map 54 × 46, with 43 VP soldiers, and 43 HP units; and (3) a map 50 × 28, with 48 VP soldiers, and 53 HP units. Algorithm 1 was executed for a value of $\wp = 20$ (i.e., 20 different games were sequentially played), and the RBP was also executed 20 times; Table 1 shows the results obtained.

Table 1. Results: VP_{win} = number of virtual player's victories, HP_{win} = number of human player's victories, $\overline{\mathrm{HP}_{death}}$ = average number of deaths in the HP army, $\overline{\mathrm{VP}_{death}}$ = average number of deaths in the VP army, $\overline{mov}$ = average number of movements, and $\overline{time}$ = average time (minutes) dedicated per game

		VP_{win}	HP_{win}	$\overline{\mathrm{HP}_{death}}$	$\overline{\mathrm{VP}_{death}}$	$\overline{mov}$	$\overline{time}$
$map_{50\times50}$	RBP	4	16	6	7	5345	3.56
	PMEA	6	14	7	7	4866	3.20
$map_{54\times46}$	RBP	9	11	4	3	7185	4.79
	PMEA	7	13	6	7	5685	3.80
$map_{50\times28}$	RBP	3	17	3	2	6946	4.63
	PMEA	6	14	7	6	6056	3.78

Even though in two of the three scenarios PMEA behaves better than RBP, note that no significant differences are shown; this is however an expected result as we have considered just one player what means that the player models obtained in-between two games are likely similar and thus their corresponding virtual players also are. In any case, this demonstrates that our approach is feasible as it produces virtual players comparable - and sometimes better - to specific and specialized pre-programmed scripts.

5 Conclusions

We have described an algorithm to design automatically strategies exhibiting emergent behaviors that adapt to the user skills in a one-player war real time strategy game (wRTS); to do so, a model to mimic how the human player acts in the game is first constructed during the game, and further a strategy for a virtual player is evolved (in between games) via an evolutionary algorithm.

Our proposal was compared with an expert system designed specifically for the game. Whereas no significance differences have been highlighted in the experiments, we make note that our approach has evident advantages compared to classical manufactured scripts (i.e., expert systems) used in videogame industry: for instance, it avoids the predictability of actions to be executed by the virtual player and thus guarantees to maintain the interest of the player This is specially interesting when the game involves more than one player as our approach would allow to construct virtual players adapted particularly to each of the human players (and this cannot be obtained with a pre-programmed script).

Further research will cope with multi-player games and thus multi-objective evolutionary programming techniques should be considered.

Acknowledgements. This work is supported by project NEMESIS (TIN-2008-05941) of the Spanish Ministerio de Ciencia e Innovación, and project TIC-6083 of Junta de Andalucía.

References

1. Lidén, L.: Artificial stupidity: The art of intentional mistakes. In: AI Game Programming Wisdom 2, pp. 41–48. Charles River Media, Inc. (2004)
2. Ahlquist, J.B., Novak, J.: Game Artificial Intelligence. Game Development essentials. Thomson Delmar Learning, Canada (2008)
3. Buro, M.: Call for AI research in RTS games. In: Fu, D., Orkin, J. (eds.) AAAI workshop on Challenges in Game AI, San Jose, pp. 139–141 (2004)
4. Corruble, V., Madeira, C.A.G., Ramalho, G.: Steps toward building of a good ai for complex wargame-type simulation games. In: Mehdi, Q.H., Gough, N.E. (eds.) 3rd International Conference on Intelligent Games and Simulation (GAME-ON 2002), London, UK (2002)
5. Forbus, K.D., Mahoney, J.V., Dill, K.: How qualitative spatial reasoning can improve strategy game ais. IEEE Intelligent Systems 17(4), 25–30 (2002)
6. Louis, S.J., Miles, C.: Playing to learn: case-injected genetic algorithms for learning to play computer games. IEEE Trans. Evol. Comput. 9(6), 669–681 (2005)
7. Stanley, K.O., Bryant, B.D., Miikkulainen, R.: Real-time neuroevolution in the nero video game. IEEE Trans. Evol. Comput. 9(6), 653–668 (2005)
8. Livingstone, D.: Coevolution in hierarchical ai for strategy games. In: IEEE Symposium on Computational Intelligence and Games (CIG 2005), Essex, UK, IEEE, Los Alamitos (2005)
9. Miles, C., Louis, S.J.: Co-evolving real-time strategy game playing influence map trees with genetic algorithms. In:International Congress on Evolutionary Computation, Portland, Oregon. IEEE press, New York (2006)
10. Lichocki, P., Krawiec, K., Jaśkowski, W.: Evolving teams of cooperating agents for real-time strategy game. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., Machado, P. (eds.) EvoWorkshops 2009. LNCS, vol. 5484, pp. 333–342. Springer, Heidelberg (2009)

11. Beume, N.,et al.: Intelligent anti-grouping in real-time strategy games. In: International Symposium on Computational Intelligence in Games, Perth, Australia, pp. 63–70 (2008)
12. Keaveney, D., O'Riordan, C.: Evolving robust strategies for an abstract real-time strategy game. In: International Symposium on Computational Intelligence in Games, Milano. Italy, pp. 371–378. IEEE press, New York (2009)
13. Hagelbäck, J., Johansson, S.J.: A multi-agent potential field-based bot for a full RTS game scenario. In: Darken, C., Youngblood, G.M. (eds.) Proc. Fifth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2009), Stanford, California, USA, The AAAI Press, Menlo Park (2009)
14. Sweetser, P.: Emergence in Games. Game development. Charles River Media, Boston (2008)

Decision Tree-Based Algorithms for Implementing Bot AI in UT2004

Antonio J. Fernández Leiva and Jorge L. O'Valle Barragán

Dept. Lenguajes y Ciencias de la Computación, ETSI Informática,
Campus de Teatinos, Universidad de Málaga,
29071 Málaga – Spain
afdez@lcc.uma.es, jlobarragan@gmail.com

Abstract. This paper describes two different decision tree-based approaches to obtain strategies that control the behavior of bots in the context of the Unreal Tournament 2004. The first approach follows the traditional process existing in commercial videogames to program the game artificial intelligence (AI), that is to say, it consists of coding the strategy manually according to the AI programmer's experience with the aim of increasing player satisfaction. The second approach is based on evolutionary programming techniques and has the objective of automatically generating the game AI. An experimental analysis is conducted in order to evaluate the quality of our proposals. This analysis is executed on the basis of two fitness functions that were defined intuitively to provide entertainment to the player. Finally a comparison between the two approaches is done following the subjective evaluation principles imposed by the "2k bot prize" competition.

1 Introduction and Related Work

Until recently, research on videogames was mainly focused on having more realistic games by improving graphics and sound. However, in recent years, hardware components have experienced exponential growth and players demand higher quality opponents controlled by better artificial intelligences (AI). In this context AI plays an important role in the success or failure of a game and some major AI techniques have already been used in existing videogames [1,2] (e.g., evolutionary computation and neural networks are beginning to be considered with moderate success [3]). However, traditionally game developers have preferred standard AI techniques such as Artificial Life, Neural Networks, Finite State Machines, Fuzzy Logic, Learning, and Expert Systems, among others [4,5]. One technique used with success for implementing the game AI in First Person Shooter (FPS) games is Decision Tree (DT); a decision tree is basically a tree that takes as input a specific situation (e.g., a combination of values corresponding to a set of perceptions) and outputs a yes/no decision. In other words, a decision tree is a tree in which its internal nodes represent questions that can be answered with yes or no, and its leaves are actions to be executed. Decision trees have already been employed as decision-making techniques in successful AAA

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 383–392, 2011.

videogames (e.g., Crysis). The reasons for their use are clear: DTs are based on a simple concept, can be easily interpreted, and helps to smooth the complexity of other techniques (e.g., finite state machines are also easy to understand and be interpreted but they introduce a high complexity if the model involves a high number of states as it is not always easy to manage the whole set of relations among all the states).

In FPS games, requiring higher quality opponents means obtaining enemies exhibiting intelligent behavior; however, it is not easy to evaluate what a 'human-like intelligence' means for a bot in these games. Generally speaking, it is well known that the Turing Test is a procedure proposed by Alan Turing to corroborate the existence of intelligence in a machine [6]. The basic fundament is that a machine that behaves intelligently might be considered as intelligent in the human sense. In this context, the "2k bot prize" is a competition[1] that proposes an interesting adaptation of the Turing test in the context of the Unreal Tournament 2004 (UT2004), a multi-player online FPS game in which enemy bots are controlled by some kind of game AI.

This paper describes an on-going work to provide intelligence to the bots in the context of UT2004, and we propose here two different DT-based algorithms. These two different approaches are compared experimentally, first from an objective point of view considering two fitness functions, and second from a subjective point of view according to the "2k bot prize" competition.

2 Proposals to Control the Bot AI

We have tested two different ways to generate the bot AI in UT2004. The first implementation is manufactured via decision trees and was manually coded following our intuition under a process of trial, error, and debugging. In fact, this is basically the process followed in most of the existing commercial FPS games. The second of our proposals consists of automating the process of generating an adequate strategy for the bot AI and is based on genetic programming (GP) techniques [7]. In the following we describe both proposals.

2.1 A Hand-Made Decision Tree-Based AI

In a high level of abstraction the logic of the bot was manually coded as machine with the following 4 states:

- Combat (if the bot is under attack).
- Pick items (if the bot detects an item).
- Pursue (if the bot detects an enemy).
- Idle (otherwise).

Each of these states is implemented with a specific hand-coded binary DT (i.e. internal nodes have two children at most). Internal nodes represent bot perceptions that involve a question with two possible answers: yes or no, and leaves

[1] http://botprize.org/ (accessed 14th of February, 2011).

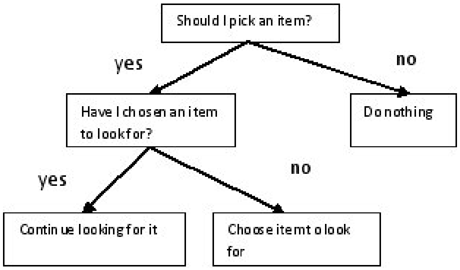


Fig. 1. Simplified decision tree for 'Pick Item'

contain actions that have to be directly executed by the bot; note that a specific action associated to a leaf is executed when all the internal nodes in the path from the tree root to the leaf have been traversed. The perceptions that the bot perceives are of the type "have I an item close?", "am I under attack?", "do I see an enemy?", "am I armed?", "do I see an item?", etc. The actions that a bot can execute are of the type "shoot", "do nothing", "re-equip with weapons", "look for a specific item (i.e., health kit, weapon, armor,.,etc.)", "pick item", "turn (right, left)", "jump",. . . ,etc. Due to the complexity of these trees we will not show them completely. Figure 1 shows a (very simplified) illustration of the tree associated to state "Pick Item" (here the perception "Have I chosen an item to look for?", and the actions "Continue looking for it", and "Choose item to look for" are over-simplifications corresponding to more complex DTs.

2.2 A Genetic Programming Based Approach

Note that the approach previously described requires an important effort from the programmer point of view as the strategy that controls the behavior (i.e., the logic) of the bot has to be intuitively coded. In fact, this strategy was the result of directly applying our experience as players. We have also considered an automated process to generate bot AI. This process is based on genetic programming (GP) techniques [7]. More specifically, we have implemented a GP algorithm to optimize the bot AI. The basic idea here is that the individuals are represented as decision trees (in fact as 4-states machines as explained in the previous section) where the actions correspond directly with terminal symbols and the perceptions are associated with non-terminal symbols. Our automated algorithm is a standard GP method that uses the following parameters: random initialization via the classical "grow" method, binary tournament selection, crossover based on random node selection and interchange of the respective subtrees, mutation based on the replacement of a complete subtree, and replacement of the worst parent. Our GP algorithm manages the following set of symbols:

T(Terminals) = Jump, RunToTarget, RunAway, Attack, Shoot, StopShooting, Disarm, Arm, TurnSomeDegrees, GoToPosition, Idle, PickWeapon, PickMedicalKit, PickArmor, PickAmmunition, NoOperation.

NT(Non-terminals) = ClosestWeapon,ClosestArmor, Distance, ClosestEnemy, ClosestMedicalKit, ClosestAmmunition, SeeEnemy, VisibleObject, WeaponLoaded, AmIShooting, WeaponType, AmIUnderAttack, AmmunitionAmount, HealthAmount.

The meaning of the symbols should be clear from their terms and we will not describe them in detail. To avoid the well-known problem of bloat in GP a maximum depth (i.e., 15 levels) was imposed for our decision trees.

3 Experimental Analysis

In this section we show the results obtained experimentally by considering separately the two proposals described previously and execute a comparative study in the context of the "2K bot prize" competition. The construction of the game AI was done for the videogame UT2004 and, although this provides its own programming language UnrealScript, our programs were coded in Java.

3.1 General Issues

In the experiments we have considered as opponent one of the bots predefined in Pogamut (i.e., a plug-in for Netbeans that is a free toolkit for development of bots in Unreal Tournament 2004); more specifically we have considered the hunter one. This election was based on our experience as players and with the aim of reaching a more exciting gaming experience. The hunter bot was tested against each of our proposals separately in a game. This experiment (i.e., this game) was executed 5 times in 5 different scenarios (i.e., maps). Each game was executed during 1 minute and a half (note that each game was executed in real time and this was very time-consuming as unfortunately we couldn't ascertain how to execute the game off-line).

Two different variants of the fitness function where considered:

$$\text{(A more aggressive function) } f_1(x) = (id + 50 * \text{frags}) - \tfrac{rd+50*d}{2}$$

$$\text{(A less aggressive function) } f_2(x) = \tfrac{id+50*\text{frags}}{2} - (rd + 50 * d)$$

Here x represents a candidate solution, 'id' (resp. 'rd') is the damage inflicted (resp. received) by our bot to (resp. from) the opponent bot (in terms of health units), 'frags' is the number of enemies annihilated by our bot, and 'd' is the number of times that our bot was destroyed by the built-in bot. These definitions for the fitness were obtained according to our intuition and based on our gaming experience as players of FPS games.

3.2 The Hand Made Strategy

Our first proposal based on the direct encoding of decision trees is referred here as 'Hand Made bot' (HM). 5 different maps (i.e., TrainingDay, Sulphur, etc and

Table 1. Summary with different fitness strategies

Fitness	Training Day	Dodging Map	Sulphur	Metallurgy	Morpheus	Average
+aggressive						
Hunter/Best	0.8205	0.8120	0.1045	0.3130	0.2280	0.4556
Hunter/Worst	0.4590	0.0000	0.0000	0.0000	0.0000	0.0918
HandMadeBot/Best	0.5535	1.0000	0.9280	0.3000	0.4125	0.6388
HandMadeBot/Worst	0.0000	0.2540	0.0000	0.0780	0.0000	0.0664
−aggressive						
Hunter/Best	0.9150	0.7115	0.3285	0.3250	0.4700	0.5500
Hunter/Worst	0.3780	0.2600	0.0025	0.0000	0.1065	0.1494
HandMadeBot/Best	1.0000	0.6405	0.6335	0.8215	0.9075	0.8006
HandMadeBot/Worst	0.4395	0.1370	0.1200	0.2970	0.3545	0.2696

5 different executions on each of them were done) and Table 1 shows the best and average values obtained by our bot as well as the opponent one; the two different fitness functions mentioned above were also considered. Observe that the HM behaves acceptably well and beats the hunter bot in most of the games.

We have also evaluated our hand-made proposal according to the bases of the "2k bot prize", that is to say, via a subjective evaluation involving a number of human judges. To do so, we have asked five different persons (i.e., undergraduate students that wanted to participate in the experience) to evaluate which of the bots behaved in a more human-like way (even though, as mentioned in the introduction section, it is not easy to define what this means). In this experiment we considered our HM bot facing other two different bots (i.e., the built-in 'hunter' and 'simple' bots). The judges were informed about the rules of the "2k bot prize" competition and were said that at least one human was playing (i.e., they were misled[2]). The mission of each judge was to select the bot with a 'more human-like behavior'. Results are shown in Table 2 in which 'B' denotes 'hunter' or 'simple bot' indistinctly. We can observe that HM is chosen by a majority of the judges. The main reason for it might be that our HM bot exhibits a less mechanical behavior than those of the predefined bots. This is not surprising as HM is based on four complex decision trees (one for each of the states mentioned previously) that consider more combinations of perceived values than the predefined bots.

3.3 The GP Approach

We have also executed our GP based algorithm in the map TrainingDay. 16 runs were done with the following parameters: number of generations: 20, Offspring size: 10, crossover probability: 0.8, mutation probability: 0.1, and fitness function: the less aggressive one. Initial population was randomly generated (a

[2] Note that these experiments are not really like in the "2k bot prize" competition since no human players are on the game.

Table 2. Subjective evaluation according to the "2K bot prize" rules; 'B' denotes predefined 'hunter' or 'simple' bot indistinctly

	Round 1	Round 2	Round 3	Round 4	Round 5
Judge 1	HM	HM	B	HM	HM
Judge 2	HM	B	B	HM	HM
Judge 3	HM	HM	HM	HM	HM
Judge 4	B	B	HM	B	B
Judge 5	HM	HM	B	B	HM
Total HM	4	3	2	3	4
Total B	1	2	3	2	1

test to avoid non-factible candidates was also used). The reason because we only consideres one scenario is because this algorithm is time-consuming as the candidate evaluation has to be done in real time; note that in each generation 10 new individuals are produced and this means to dedicate 15 minutes per evaluation in each generation (i.e., globally this algorithm takes 5 hours). Precisely for this reason no tuning of the parameters were done.

Table 3. Improvement percentage: 16 runs; GP approach

Run	Worst	Best	% improvement	Run	Worst	Best	% Improvement
1	0,0500	0,2335	18,35%	9	0,1242	0,5568	43,26%
2	0,2755	0,3155	4,00%	10	0,2885	0,2885	0,00%
3	0,6375	0,6375	0,00%	11	0,1650	0,1650	0,00%
4	0,0950	0,1570	6,20%	12	0,0820	0,1475	6,55%
5	0,2671	0,7786	51,15%	13	0,0350	0,3805	34,55%
6	0,2065	0,2065	0,00 %	14	0,3970	0,4485	5,15%
7	0,1400	0,5926	45,26%	15	0,0000	0,6790	67,90%
8	0,2345	0,6323	39,78%	16	0,0070	0,4172	41,02%

Table 3 displays the improvement percentage that this algorithm provided in each of the 16th runs. The *improvement percentage* is measured as the difference in fitness values between the best initial random candidate generated in the initial population and the best final solution. Several considerations can be done here: (1) the improvement seems to be not appreciable if the initial solution is acceptably good (as for instance in runs 3, or 6); (2) the best overall solution obtained has a fitness value of 0.7786 (see run 5) and was evolved from an initial value of 0.2671 representing an improvement of about 52% (Figure 2 illustrates this evolution). Note however that this solution is worst (according to its fitness value) than the best solution obtained by the HM bot (i.e., 1.000); initially this might indicate a clear superiority of HM with respect to the GP algorithm. However this does not necessarily mean that HM behaves in a more human-like way than the evolved bot. The next section tries to shed some light on this issue.

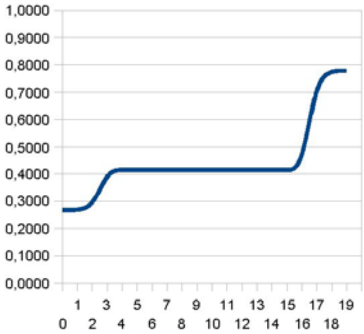


Fig. 2. Evolution of fitness in the 5th run of the GP algorithm. map = TrainingDay

Table 4. Subjective evaluation according to the “2K bot prize” rules; EV = evolved bot via GP algorithm; HM = Hand made bot

	Round 1	Round 2	Round 3	Round 4	Round 5
Judge 1	EV	HM	HM	HM	EV
Judge 2	EV	EV	EV	HM	HM
Judge 3	EV	HM	EV	EV	EV
Judge 4	HM	EV	EV	EV	EV
Judge 5	EV	EV	HM	HM	EV
Total HM	1	2	2	3	1
Total EV	4	3	3	2	4

3.4 A Subjective Comparison between the Two Proposals

We have conducted a comparison between the HM bot and the best solution (denoted in the following as EV = ‘evolved bot’) obtained with our (non-optimized) GP algorithm according to the rules imposed in the “2k bot prize” competition. To do so, we have again requested 5 judges (i.e., undergraduate students) to evaluate which of these two bots (i.e., the HM and the EV) behaves in a more human-like way and the results are illustrated in Table 4. Surprisingly, EV was the best evaluated in this subjective process. One reason to justify this fact might be that a bot AI that a priori is not too good surely behaves erroneously in certain situations and this represents a more human-like behavior with respect to an optimal strategy. In any case, this result should be taken carefully again as this evaluation is totally subjective and no empirical data were analyzed. This still remains as an issue of further work.

4 Related Work

Evolutionary algorithms (EAs) (we use this term in a broad sense to refer to any kind of evolutionary procedure, including genetic algorithms and genetic programming) offer interesting opportunities for creating intelligence in strategy or role-playing games and, on the Internet, it is possible to find a number of articles related to the use of evolutionary techniques in videogames. For instance, [8] shows how EAs can be used in games for solving pathfinding problems; also [3] focused on bot navigation (i.e., exploration and obstacle avoidance) and proposed the use of evolutionary techniques to evolve control sequences for game agents. However, till recently most of the work published on the use of EAs in games was aimed at showing the important role EAs play in Game Theory and, particularly, their use in solving decision-taking (mainly board) games [9,10,11,12,13,14].

Evolutionary techniques involve considerable computational cost and thus are rarely used in on-line games. One exception however, published in [15], describes how to implement a genetic algorithm used on-line in an action game (i.e. a first/third person shooter). In fact, the most successful proposals for using EAs in games correspond to off-line applications, that is to say, the EA works on the user's computer (e.g., to improve the operational rules that guide the opponent's actions) while the game is not being played and the results (e.g., improvements) can be used later online (i.e., during the playing of the game). Through offline evolutionary learning, the quality of opponent intelligence in commercial games can be improved, and this has been proven to be more effective than opponent-based scripts [16].

Related with the work described here, [17] applies GP techniques to evolve bot AI in Unreal$^{\mathrm{TM}}$. Although there are similarities with our second proposal described here, there are also evident differences: the main distinction is that bot AI is coded in [17] as finite state machines (FSM) that do not match our decision trees (even though these FSM were internally coded as trees); also only 2 states were considered and the rules governing the bot AI just admitted two inputs (translated to our work described here this would represent decision trees with only two levels of depths). Moreover, a 8-players game was used for the experimentation whereas we have considered a version of two/three players; in addition no subjective evaluation was considered and the fitness function was different to ours. In any case, the results obtained in [17] support our conclusions that GP is a promising approach to evolve bot AI in FPS games. The same authors also explored the employment of genetic algorithms to controlling bots in FPS games [18].

5 Conclusions and Further Work

This paper has dealt with the problem of providing artificial intelligence to non-player characters (i.e., the bots) in first person shooter games, and more specifically in the context of the game Unreal Tournament 2004. Two different proposals based on Decision Trees to code bot AI in UT2004 have been described. The first proposal represents the approach that currently is followed in the development of existing commercial games and consists of manually coding the bot AI. This way, our manufactured bot AI has been pre-programmed as a decision tree with multiple rules. The second approach is based on genetic programming, and consists of evolving automatically with no human intervention) a set of candidate solutions represented as decision trees.

We have also conducted an experimental analysis and have compared our two proposals, firstly from an objective point of view according to two specific fitness functions (defined from our gaming experience), and second from a subjective point of view according to the basis of the “2k bot prize” competition. Our two proposals have both advantages and drawbacks. While it is evident that game industry is demanding automated processes to automatically generate the so-called game AI, it is also clear that this is a difficult task as it is to be very dependant on the evaluation phase (and more specifically on the definition of the fitness function). The difficulty of establishing a good fitness function in FPS games comes from considering the user satisfaction as the value to optimize and this represents a ‘hard’ obstacle as this component is not easy to quantify mathematically; there exist however some interesting papers that open interesting research lines - see for instance [19,20].

In this paper our hand-coded strategy provides better fitness values than the best evolved strategy generated automatically via a standard genetic programming (GP) algorithm. In addition the GP algorithm is time-consuming as the evaluation function requires simulation in real time. However, there are other issues that favor the automated process. For instance, the hand-coding is also a costly process that requires many hours of coding; in addition this process of trial, error and debugging is also very expensive measured in human resources. Surprisingly the evolved strategy was better evaluated than the hand-coded one when we considered a subjective evaluation close to that proposed in the “2k bot prize” competition. A more detailed analysis on this issue should be conducted in the future although we have already pointed out some reason for it.

As further research we plan to optimize our GP algorithm by a simple tuning of its parameters. Also, alternative (and more complex) definitions for the fitness functions, that take into account user’s satisfaction, surely would improve the evolution of strategies. In this sense, human-guided interactive evaluation might be useful and thus interactive optimization will be considered in the future. It would be also interesting to see how the two approaches work in a larger scale experiment.

Acknowledgements. This work is supported by project NEMESIS (TIN-2008-05941) of the Spanish Ministerio de Ciencia e Innovación, and project TIC-6083 of Junta de Andalucía.

References

1. Johnson, D., Wiles, J.: Computer games with intelligence. In: FUZZ-IEEE, pp. 1355–1358 (2001)
2. Millington, I.: Artificial Intelligence for Games. In: Interactive 3D Technology. Morgan Kaufmann, San Francisco (2006)
3. Buckland, M.: AI Techniques for Game Programming. Premier Press (2002)
4. Bourg, D., Seemann, G.: AI for Game Developers. O’Reilly, Sebastopol (2004)

5. Miikkulainen, R., Bryant, B.D., Cornelius, R., Karpov, I.V., Stanley, K.O., Yong, C.H.: Computational intelligence in games. In: Computational Intelligence: Principles and Practice, pp. 155–191. IEEE Computational Intelligence Society, Piscataway (2006)
6. Turing, A.: Computing machinery and intelligence. Mind 59, 433–460 (1950)
7. Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
8. Sweetser, P.: How to build evolutionary algorithms for games. In: AI Game Programming Wisdom 2, pp. 627–638. Charles River Media, Inc. (2003)
9. Kim, K.J., Cho, S.-B.: Evolutionary algorithms for board game players with domain knowledge. In: Baba, N., Jain, L.C., Handa, H. (eds.) Advanced Intelligent Paradigms in Computer Games, pp. 71–89. Springer, Heidelberg (2007)
10. Fogel, D.B.: Evolving a checkers player without relying on human experience. Intelligence 11(2), 20–27 (2000)
11. Chellapilla, K., Fogel, D.B.: Evolving an expert checkers playing program without using human expertise. IEEE Trans. Evolut. Comput. 5(4), 422–428 (2001)
12. Wittkamp, M., Barone, L.: Evolving adaptive play for the game of spoof using genetic programming. In: Louis, S.J., Kendall, G. (eds.) IEEE Symposium on Computational Intelligence and Games (CIG 2006), University of Nevada, Reno, campus in Reno/Lake Tahoe, pp. 164–172. IEEE, Los Alamitos (2006)
13. Pollack, J.B., Blair, A.D.: Co-evolution in the successful learning of backgammon strategy. Machine Learning 32(3), 225–240 (1998)
14. Ong, C., Quek, H., Tan, K., Tay, A.: Discovering chinese chess strategies through coevolutionary approaches. In: IEEE Symposium on Computational Intelligence and Games (CIG 2007), pp. 360–367. IEEE, Los Alamitos (2007)
15. Fernández, A.J., Jiménez, J.G.: Action games: Evolutive experiences. In: Reusch, B. (ed.) Computational Intelligence, Theory and Applications, International Conference 8th Fuzzy Days. AISC, vol. 33, pp. 487–501. Springer, Heidelberg (2004)
16. Spronck, P., Sprinkhuizen-Kuyper, I., Postma, E.: Improving opponent intelligence through offline evolutionary learning. International Journal of Intelligent Games & Simulation 2(1), 20–27 (2003)
17. Mora, A.M., Montoya, R., Guervós, J.J.M., Sánchez, P.G., Castillo, P.Á., Laredo, J.L.J., García, A.I.M., Espacia, A.: Evolving bot AI in unreal^{tm}. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcazar, A.I., Goh, C.-K., Merelo, J.J., Neri, F., Preuß, M., Togelius, J., Yannakakis, G.N. (eds.) EvoApplicatons 2010. LNCS, vol. 6024, pp. 171–180. Springer, Heidelberg (2010)
18. Esparcia-Alcázar, A.I., García, A.I.M., Mora, A., Guervós, J.J.M., García-Sánchez, P.: Controlling bots in a first person shooter game using genetic algorithms. In: CEC 2010, Barcelona, Spain, pp. 1–8. IEEE, Los Alamitos (2010)
19. Andrade, G., Ramalho, G., Gomes, A., Corruble, V.: Dynamic game balancing: an evaluation of user satisfaction. In: AIIDE Artificial Intelligence and Interactive Digital Entertainment, pp. 3–8. AAI Press (2006)
20. Yannakakis, G.N.: How to model and augment player satisfaction: A review. In: 1st Workshop on Child, Computer and Interaction, Crete. ACM Press, New York (2008) (Invited paper)

Neural Networks versus Genetic Algorithms as Medical Classifiers

Oscar Marín, Irene Pérez, Daniel Ruiz, Antonio Soriano, and Joaquin D. García

Department of Computer Technology. University of Alicante, PO 99,
03080 Alicante, Spain
{omarin,iperez,druiz,soriano,jdgarcia}@dtic.ua.es

Abstract. In this article we want to assess the feasibility of using genetic algorithms as classifiers that could be used in clinical decision support systems, for urological diseases diagnosis in our case. The use of artificial neural networks is more common in this field, and we have previously tested their use with the same purpose. At the end of the document we compare the obtained results using genetic algorithms and two different artificial neural networks implementations. The obtained accuracy rates show that genetic algorithms could be a useful tool to be used in the clinical decision support systems field.

Keywords: neural networks, genetic algorithms, clinical decision support systems, urological disorders.

1 Introduction

Doctors often use as a basis to diagnose their expert knowledge, the responses obtained from patients to questions about their illnesses and the results of relevant clinical tests. This is a long and complex process which has no mathematical precision and may be influenced by many external factors, which can impede the experts to take the right decision about a final diagnosis and the possible treatments to be performed. An example could be the case of a patient that may suffer from various diseases, which makes him to give contradictory and unclear answers to doctor's questions about his symptons. Other scenario could be the case of a patient who has some clear ailments but the real cause of these is an undetected problem; this fact would delay the start of an appropiate treatment [1]. Also in situations of shortage of doctors, especially in rural areas, a clinical decision support system could be used for prevention and early detection of health problems [2].

In our case, we have used as basis of this article a database of tests performed over patients with urological disorders. In this case, the knowledge about the origin of the detected dysfunctions depends largely on the experience of the experts, and the research activity that they constantly perform. Specifically, in urology field there are many disorders which exact diagnosis is difficult to do, due to the interaction with the neural system and the limited knowledge we have about it. Within this environment, and based on machine learning theory, clinical decision support

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 393–400, 2011.

systems (CDSS) emerge to support the experts' decision making tasks [2]. Using a diagnostic support system could reduce the number of needed tests having as consequences the improvement of the global patient care and the reduction of the costs associated with urodynamic testings. For these reasons and many others medical decision support systems are used more and more.

A key question to build up these systems is to choose the suitable classifier to perform the diagnostic process. In this study we want to test if genetic algorithms are a good alternative to other strategies commonly used in CDSSs. To do this, a comparison between the diagnostic results of two implemented ANNs and a genetic algorithm will be done.

2 Automatic Classification Techniques and Diagnosis

Since its rising, the field of machine learning theory has had as one of its main objectives the use of its techniques in the field of health [3] [4].

This area of knowledge describes algorithms and techniques that have the aim of solving problems related to the automatic acquisition of knowledge, in order to simulate the trial of an expert in solving problems of classification and discrimination. This is the aspect of machine learning that is most interesting for the healthcare field: the possibility of using its techniques for classifying symptoms, diseases,..., to automatically get a diagnosis as if it were a human expert.

Within machine learning, artificial neural networks (ANNs) are not the only, but an extensively used tool to perform automatic classification tasks. Many different kinds of ANNs have been widely used in the diagnose of several diseases,for instance, multi-layer perceptrons and radial basis function neural networks [5] [6], self-organized maps [7], and others [8].

Besides ANNs, there are other more recent soft computing strategies like genetic algorithms (GA), but their application in medical enviroments is much lower than in ANNs case. Even, in most cases GAs are not used for diagnostic purposes, and only to optimize the feature selection process. This lead us to test the feasibility of using GAs as a basis of a diagnostic algorithm comparing its performance with other widely proofed classifiers in medical diagnosis, like multi-layer perceptrons.

2.1 Genetic Algorithms for Diagnosis

Genetic algorithms are included in the area of artificial intelligence, specifically inside evolutionary computing. They are so named because they are inspired by biological evolution and molecular-genetic basis [9]. To use a genetic algorithm, first we must define a population to work on. Each individual of this population would be what we call chromosome and each chromosome consists of a string of features each of which we call gene. When the chains of chromosomes are represented in binary form, we call it genotype. After defining the population, it evolves by means of interactions among chromosomes. This will produce new generations of chromosomes, and we call each new generation, descendants.

The initial population is randomly generated to ensure the diversity of chromosomes and represent the majority of the population. As in nature, not all chromosomes are capable of reproduction. This will be decided by applying over them a function called fitness, which assigns a score to each chromosome depending on how close they are to the best possible solution.

There are several ways to get the selection of the next generation, the main ones are the crossover, and mutation. In the crossover two chromosomes are randomly selected to combine their genes and obtain two new descendant chromosomes, composed of the parental characteristics combination. This is the main operation in genetic algorithms. On the other hand, there is the mutation operator that is far less common. This operator randomly selects some genes from one chromosome and changes their values with their complementaries. For example, if we were talking about a genotype, if the gene value is 0 mutation will change it to 1.

As we can see, genetic algorithms are systematic methods for solving search and optimization problems, by means of applying to them the same methods of biological evolution: population-based selection, reproduction and optimizing mutation. We can find some researches focused in the application of genetic algorithms to the general problem of classification [9], but very few researchers have used this soft computing tool for medical diagnosis [10]. Nevertheless, recent innovations have introduced new algorithms that overcome traditional methods and are more likely to be accepted by the medical community, which is an important fact, since the lack of acceptance of this tools by healthcare professionals is one of the most critical problems hindering the development of automatic diagnostic systems.

In 1998 it was made one of the first research projects where genetic algorithms were used in medicine known as Galactica. In this project, a learning system that used GA to discover knowledge rules from a clinical database was developed. This expert system used precise rules for the diagnosis of female urinary incontinence [11]. Subsequently, GA have also been used for the diagnosis of heart diseases using samples from tests' results stored in a huge database [1]. More recent examples of GA used in clinical field are the diagnosis of breast cancer through processing 3D images [12], and to show if there is a relationship between DNA and breast cancer [13].

2.2 Neural Networks for Diagnosis

Neural networks are not the only method of machine learning used for automatic classification tasks, but rather a widely used tool for this purpose [14]. In general, an ANN is able to model complex biological systems, revealing relationships between the input information that cannot always be recognized by conventional analysis.

To have a set of examples representing previous experiences is essential to build an ANN-based classifier that ensures a good performace of learning and generalization processes. These previous experiences are the ANNs design inputs.

After applying to these values a collection of mathematical functions at different stages, an output value is given. This output value has a useful meaning to assign a class to an input sample (e.g. positive or negative for cancer), or express the probability for this sample of being a member of a class. A classification of artificial neural networks based on the kind of learning approach followed in their design, divides them into two types: supervised and unsupervised networks [15]. The supervised networks need a supervision of the automatic training process, which specifies the corresponding output to each training input. In the case of unsupervised networks, it is expected that the network classifies the inputs into different groups according to the outputs, but in this case it is not explicitly specified to them the relationship between input and output. In this document, we will work with two different supervised ANNs: multi-layer perceptrons (MLPs) and radial basis functions neural networks (for short RBFs).

In the case of the MLP implementation, it is mainly used the backpropagation training algorithm to carry out this process [15]. MLPs have an input and an output neuron layer, and a non-fixed number of intermediate, or hidden, layers with an arbitrary number o neurons in each one.

In the case of RBF's, its structure is quite similar to multi-layer perceptron, with an input and output layer respectively. However there is a difference in the number of intermediate, or hidden, layers. RBFs have only one hidden layer while MLPs could have more than one. In addition to this, another important difference between MLPs and RBFs is the transference function that each neuron applies to the data that pass through them. In the case of MLPs it is a linear function, in most cases a sigmoid, and in the RBFs case in the intermediate layer it is radial function, mostly any kind of gaussian function, and in the output layer a linear function too.

3 Experimentation

Next, we explain the steps followed in the design, implementation and testing processes of each of the automatic classifiers. It is also presented the clinical tests database used and the preprocessing data tasks that have been carried out. Finally, we explain the results obtained from the test phase for each ANN and the GA implementation, comparing them in terms of accuracy rates.

3.1 Urological Tests Database

The database used to carry out the tests consists of samples of patients with urological problems. Our objective has been to diagnose the two main diseases that these patients suffer from. All along the experimentation we were advised by medical specialists, who helped us to have a clear idea about the meaning of the data included in the clinical tests.

Our database consists of 160 samples of patients, suffering 74 of them from incontinence and 86 from urinary tract obstruction. Each sample consists of patient's personal information and values related to 15 tests made to each patient: age, gender, urine volume, maximun and average urine flow, vesical capacity, micturition time, micturition urine volume, detrusor muscle contraction degree, residual post-micturition, first sensation, detrusor pressure at Qmax, detrusor filling pressure, lower abdominal pressure, anal tone, perianal and perineal sensitivity.

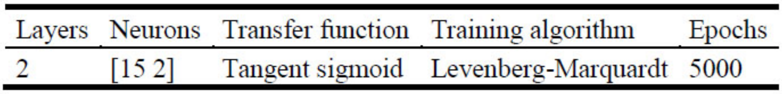
To facilitate the processing of the experimental data, we have been forced to homogenize the data in a preprocessing stage. After consulting with specialists in urology, we chose the discretization technique to do this. Since we were dealing with data from very wide ranges, as it is common in biomedicine and urology, this preprocessing strategie could be carried out with a minimum loss of information and behaves in a stable way. In the fews cases that there was loss of information, as a result of being empty any of the samples' fields, the average value of that property was assigned to those empty fields.

Moreover, given the small number of valid samples that were available for the tasks of training, validation and testing of the classifiers, we used the technique of cross validation [16], in order to have a greater number of samples to carry out these tasks. So we have divided the whole sample set into 3 disjoint groups, using one as training set and the other two as validation and test sets respectively, in several iterations. This allows us to obtain better results and better generalization rates for each classifier.

3.2 Neural Network and Genetic Algorithm Based Classifiers Implemented

The two ANNs implemented are supervised ANNs, that is, they adquire the needed knowledge to simulate real clinical experts performance, calculating the error rate obtained in partial classifications during the training stage. Deppending on the value of this error rate, a bigger or lower adjust of net's connections weights is done. As the training process goes on, the error rate is expected to be lower reducing the needed adjustments.

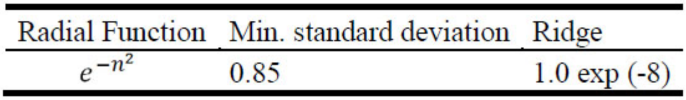
A set of training, validation and testing tests were performed using the samples obtained after applying the cross validation technique to the preprocessed input samples. Different configurations of MLPs and RBFs were tested, varying the parameter's values. For the MLP implementation we tested the use of three different transfer functions (linear, logarithmic, and tangent-sigmoid), the use of 1 or 2 hidden layers with different numbers of neurons inside them, various number of training iterations (epochs) and five modes of the backpropagation algorithm. The final values were chosen depending on the accuracy rates obtained using them. The best rate was obtained with a hidden layer of 15 neurons and an output layer of 2 neurons, where each neuron gives the probability of suffering from one of the observed diseases. The chosen training algorithm was Levenberg-Marquardt with 5000 iterations in the training stage, beyond this number of repetitions the obtained results remained the same (Fig. 1).



Layers	Neurons	Transfer function	Training algorithm	Epochs
2	[15 2]	Tangent sigmoid	Levenberg-Marquardt	5000

Fig. 1. Configuration parameters of the MLP final implementation

A similar approach was followed for the RBF implementation. In this case the tested parameters were the radial basis function to be used as transfer function in the intermediate layer, the ridge of this function and the minimun standard deviation allowed (Fig. 2).



Radial Function	Min. standard deviation	Ridge
e^{-n^2}	0.85	1.0 exp (-8)

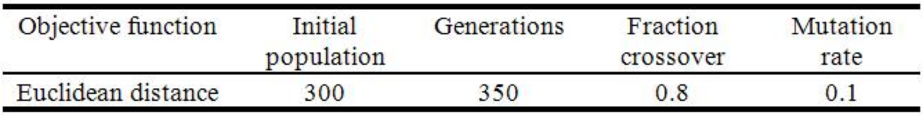
Fig. 2. Configuration parameters of the RBF final implementation

For the design and implementation of the classifier based on genetic algorithms, we tested the combination of different values for each parameter of the algorithm: initial population size, maximum number of generations to be created, target function for selecting the chromosomes, type of intersection between elite chromosomes etc.

The first step was to obtain a set of representative values of each class that could be assigned to the input samples. To do this, we generated randomly an initial population where each chromosome has an associated feature vector. Next step is selecting the elite chromosomes to do the crossing that generates the following generation. The elite chromosomes are selected using as deciding factor, the distance between their features vector and the training samples set for each class. The objective function which value we want to optimize, in this case minimize, it is a distance function, so we tested three classic distance functions such as Manhattan distance, Normal or Minkowski and Euclidean. It was the latter which gave the best results and therefore was the chosen one for the final design.

After obtaining the representative patterns of each class by means of the GA, the next step was to calculate the value of the Euclidean distance from each sample of the testing set to these class identifying patterns. Finally, it was assigned the class, i.e. the diagnostic result, of the set that was closest to each sample of the testing set, that is, the one that makes the distance function value minimum.

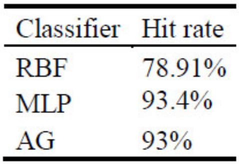
The combination of the GA configuration parameters that provides a better result in the diagnostic process, finally imlpemented, is showed below in (Fig. 3):



Objective function	Initial population	Generations	Fraction crossover	Mutation rate
Euclidean distance	300	350	0.8	0.1

Fig. 3. Configuration parameters of the GA final implementation

Finally, the results obtained after performing multiple training and testing tests using the genetic algorithm classifier and the neural networks based, have been the ones we see in(Fig. 4):



Classifier	Hit rate
RBF	78.91%
MLP	93.4%
AG	93%

Fig. 4. Obtained accuracy rates

4 Conclusions

The purpose of our study was to test the feasibility of using GAs as classifiers in a medical enviroment. Since there are other soft computing tools that have been widely tested for this purpose, as it is the case of ANNs, the comparison of the diagnostic results obtained using a GA implementatio and the results given by these other well-prooved classifiers, can be interpreted as a measure of how good the diagnostic performance of a GA implementation could be.

In our experimentation, we have designed and implemented three classifiers based on two different techniques from machine learning field. The objective was to implement an automatic system for the diagnosis of two urinary tract dysfunctions: urinary incontinence and urinary tract obstruction. To do this, we have collected data from a clinical database that contains samples from real test performed over patients diagnosed from suffering one of the two anomalies.

Two of the classifiers are based on two supervised artificial neural network models, a multi-layer perceptron and a radial basis function neural network. The third classifier was implemented using a genetic algorithm. There have been performed some tests using different parameter's configurations of each of them, to get one that provided a better accuracy rate in the diagnostic process.

Analysing the accuracy rates we can see that they were quite good in general terms of accuracy. The classifier based on a MLP gave the best results but closely followed by the one based on GA. On the other hand, despite offering good diagnostic rates, the RBF based classifier is far from being as accurate as the other classifiers.

These results show that GAs are as good as ANNs in clinical diagnostic tasks. This fact is a first step in choosing an optimal machine learning technique in order to implement a more broad and general system for diagnosing that allows us to discriminate between healthy patients and those presenting any anomaly on the urinary tract.

Future projects based on the results of this study are getting a larger number of clinical tests samples, to have a greater clinical database that could cover a wider range of urological anomalies, to test new types of machine learning techniques such as unsupervised neural networks, competitive, etc.

Acknowledgement. This work has been granted by the Ministerio de Ciencia e Innovación of the Spanish government (Ref. TIN2009-10855) and co-financed by FEDER.

References

1. Yan, H., Zheng, J., Jiang, Y., Peng, C., Xiao, S.: Selecting critical clinical features for heart diseases diagnosis with a real-coded genetic algorithm. Applied Soft Computing 8(2), 1105–1111 (2008)
2. Berner, E.S.: Clinical Decision Support Systems Theory and Practice, 2nd edn. Health Informatics Series. Springer, Heidelberg (2007)
3. Greenes, R.A.: Clinical Decision Support, The road ahead. Academic Press, London (2007)
4. Coiera, E.: Guide to Health Informatics, London, UK (2003)
5. Mehrabi, S., Maghsoudloo, M.: Application of multilayer perceptron and radial basis function neural networks in differentiating between chronic obstructive pulmonary and congestive heart failure diseases. Expert Systems with Applications (2008)
6. Zadeh, A.E., Khazaee, A., Ranaee, V.: Classification of the electrocardiogram signals using supervised classifiers and efficient features. Computer Methods and Programs in Biomedicine 99(2), 179–194 (2010)
7. Alba-Sanchez, F., Yañez-Suarez, O., Brust-Carmona, H.: Assisted diagnosis of Attention-Deficit Hyperactivity Disorder through EEG bandpower clustering with self-organizing maps. In: Procds. of the IEEE Engineering in Medicine and Biology Society (EMBC) Annual Intl. Conf., pp. 2447–2450 (2010)
8. Lisboa, P.K., Taktak, A.F.G.: The use or artificial neural networks in decision support in cancer: A systematic review. Neural Networks (2006)
9. Espejo, P.G., Ventura, S., Herrera, F.: A Survey on the Application of Genetic Programming to Classification. IEEE Transactions on Systems, Man, and Cybernetics 40(2), 121–144 (2010)
10. Yardimci, A.: Soft computing in medicine. Applied Soft Computing 9(3), 1029–1043 (2009)
11. Laurikkala, J., Juhola, M.: A genetic-based machine learning system to discover the diagnostic rules for female urinary incontinence. Computer Methods and Programs in Biomedicine 55(3), 217–228 (1998)
12. Tomar, R.S., Singh, T., Wadhwani, S., Bhadoria, S.S.: Analysis of breast cáncer using image processing techniques. In: Proceedings of the 3rd Uksim European Symposium on Computer Modeling and Simulation (2009)
13. Tseng, M., Liao, H.: The genetic algorithm for breast tumor diagnosis - The case of DNA viruses. Applied Soft Computing 9(2), 703–710 (2009)
14. Alpaydin, E.: Introduction to Machine Learning. MIT Press, Cambridge (2004)
15. Haykin, S.: Neural Networks and Learning Machines, 3rd edn. Prentice-Hall, Englewood Cliffs (2008)
16. Hu, M.Y., Zhang, G., Jiang, C.X., Patuwo, B.E.: A Cross-Validation analysis of neural network out-of-sample performance in exchange rate forecasting. Decision Sciences 30(1), 197–216 (2007)

Complexity Changes in Human Wrist Temperature Circadian Rhythms through Ageing

R. Marin[1,*], M. Campos[1], A. Gomariz[1], A. Lopez[1], M.A. Rol[2], and J.A. Madrid[2]

[1] Computer Science Faculty, University of Murcia, Spain
[2] Chronobiology Laboratory, Department of Physiology, University of Murcia, Spain

Abstract. Circadian rhythms are cycles in physiological processes that have a near-daily frequency. The wrist skin temperature has proven to be a good marker of circadian rhythmicity. In this paper we attempt to establish whether complexity changes in human circadian rhythms in ageing can be assessed through phase variability in individual wrist temperature records. To this end, we propose some phase complexity measures that are based on Lempel-Ziv complexity, Approximate Entropy, instantaneous phase, Hilbert transform and a complex continuous wavelet transform. A sample consisting of 53 healthy subjects has been studied. Our experimental results consistently show that a significant decrease in phase complexity happens when ageing.

Keywords: circadian rhythms, wrist temperature, complexity measures, age-dependent changes.

1 Introduction

Chronobiology is a scientific discipline that deals with the study of the biological rhythms and their underlying mechanisms. Biological processes that repeat themselves approximately every 24 hours are called daily or circadian rhythms. Most physiological, behavioural and biochemical variables, such as body temperature, blood pressure, sleep, feeding and neuroendocrine factors exhibit circadian rhythmicity. Circadian rhythms under artificially constant environmental conditions run with a period slightly different from 24 hour; however, under natural conditions their phase and period entrain to light dark cycle [13,20].

The circadian system of mammals is composed by a network of structures responsible for the generation and synchronization of circadian rhythms to environment. It is composed of a central pacemaker, located in the suprachiasmatic nucleus of hypothalamus (SCN) and several peripheral clocks. At molecular level, circadian clocks are composed of a set of genes and proteins that

[*] The authors wish to thank the Seneca Foundation, the Ministry of Science and Innovation and the Instituto de Salud Carlos III for their financial support of this study through the projects 08853/PI/08, TIN2009-14372-C03-01, TRA2009-0141, RETICEF (RD06/0013/0019) and BFU 2010-21945-CO1.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 401–410, 2011.

generate self-sustained circadian oscillations through positive and negative transcriptional/translational feed-back loops, being *Bmal1*, *Clock*, *Per* and *Cry* the most relevant clock genes [4,12,21].

On the other hand, thermoregulatory processes are essential in mammals. An homeostatic mechanism keeps the body temperature around a set point, that is is modulated by circadian rhythmicity. Core temperature decreases during sleep in order to slow down metabolic activity. This is accomplished by means of the loss of heat through peripheral blood vessels vasodilation, producing an increase in skin temperature. Peaks in skin temperature correspond to sleep stages. The wrist skin temperature, as measured by a wearable temperature sensor has proven to be a good marker of the circadian rhythms of the subject. The Chronobiology Laboratory of the University of Murcia designed and validated a data recording protocol that allows ambulatory assessment of human circadian rhythms by measuring wrist skin temperature [15,14,10].

Little is known about how circadian system evolves over years. The most prominent effects would be an amplitude decrease and progressive phase advances relative to the light dark cycle. Some authors [5] suggest that ageing is related to instability of circadian phase and increased day-to-day variability. However, other studies have shown a decrease [6]. Endogenous circadian rhythm seems to remain stable, but the ability to synchronize the endogenous clock with the cyclic environment seems to be impaired with age [21]. The term "robustness" is related to the regularity of circadian rhythms. Formal definition are lacking although several indexes have been proposed [12].

Physiological complexity is also an elusive but evident feature of living beings. Many biological signals exhibit a complexity loss with age. Complexity loss is related to structural and functional alterations on systems composed of many subsystems and feed-back loops [17]. No single comprehensive measure of biological complexity exists. In practice, complexity measures are based either in entropy-related concepts or in deterministic nonlinear dynamics. In cardiovascular dynamics assessment some useful measures are Approximate Entropy (*ApEn*), detrended fluctuation analysis (DFA) scaling exponent, fractal dimensions or Lyapunov exponents. Circadian studies based on a comprehensive set of complexity measures are still scarce [19,2].

In this paper we attempt to establish whether complexity changes in human circadian rhythms in ageing can be assessed through phase irregularities in individual wrist temperature records. To this end, we analyze what kind of correlation, if any, exists between age and phase complexity. We propose some phase complexity measures. A novel approach based on *LZ* complexity, instantaneous phase and Hilbert transform is introduced. The *ApEn* complexity of instantaneous phase is also studied, using both the Hilbert transform approach and a complex continuous wavelet transform (CWT). Our results consistently show that a significant decrease in phase complexity happens when ageing.

This work is part of a wider research in Ambient Assisted Living frameworks for the elderly, where detecting changes in the individual circadian pattern could be useful to enhance the reliability of system decisions and to estimate the dynamic evolution of physiological age relative to the chronological age.

The structure of the paper is as follows. Section 2 describes the experimental setup. Complexity measures are explained in section 3. Our results are presented and discussed in section 4. Finally, we summarize our conclusions in section 5.

2 Experimental Setup

2.1 Data Acquisition and Preprocessing

A sample of 126 healthy subjects (ages 9-75 yr) was initially selected. The volunteers were encouraged to maintain their usual lifestyles. A Thermochron iButtom miniaturized data logger (Dallas, Maxim) was placed on the wrist of the non-dominant hand with the help of a wristband. Temperature was logged every 10 minutes for a variable number of days. To ease the interpretation of results we will uses days (d) as time unit and cycles/day (c/d) as frequency unit. Thus, the sampling period is $T = 0.0069\,d$ and the sampling frequency is $F_s = 144\,c/d$. The signal was filtered with a 8th order Chebyshev Type I low-pass antialiasing filter with a cut-off frequency of $57.60\,c/d$ ($0.8 \times F_{Nyquist}$).

The recorded signal exhibits occasional artifacts displayed as large temperature drops, that appear when the subject removes the wristband or when the sensor is badly positioned for some time. A simple preprocessing algorithm removes large artifacts and replaces missing data with interpolate data. A reliability index was defined as the ratio between the number of replaced samples and total number of samples. Sixty records with reliability index smaller than 90% were discarded. Moreover, 13 short records (duration less than $7\,d$) were also discarded. Larger records were clipped to the seven central days. Thus, the final sample consists of 53 records with a duration of $7\,d$ (1008 samples).

2.2 Statistical Analysis and Surrogates

Most of the spectral energy of recorded signals is confined below $5.46 \pm 1.83c/d$. This was determined by applying a 32 samples median filtered periodogram. Non filtered periodogram shows a large peak frequency of 1 c/d for most of the records, corresponding to a normal entrained circadian rhythm. Previous studies reveal the existence of weak nonstationarities on circadian rhythms [12]. We conducted a study based on the application of 1st order symlets, an orthogonal discrete wavelet transform (DWT), focused on the ranges 0-1.13 and 1.13-9.02 c/d. Most of the energy ($99.95 \pm 0.05\%$) is concentrated in the first range.

A 4th order Chebyshev Type I low-pass filter with a cut-off frequency of $5.46\,c/d$ was applied to the raw signal. A different filtering approach was also tested. We applied the 5th order Daubechies DWT and reconstructed the A4 approximation. This resolution level roughly correspond to a LP cut-off frequency of 6.04 c/d. We will denote the raw signal by s, the first filtered signal by *sflp* and the wavelet reconstruction by *sfa4*.

Parametric and non-parametric methods have been applied to test the existence of a significant statistical dependence between complexity measures and age. On the one hand, the Spearman's rank correlation coefficient ρ was determined. We will denote the resulting p-value as ps. On the other hand, the coefficient of determination R^2 was computed by means of a simple first order linear regression. In this case the p-value, denoted as p, is obtained by applying an F-test. Values for ρ have to be compared with the correlation coefficient r ($r = \sqrt(R^2)$). The significance level for both cases was fixed as $\alpha = 0.01$.

A surrogate data methodology was applied to ensure reliable results. Surrogate data hypothesis testing consist on generating an ensemble of artificial random time series that preserve a given set of features of the original signal [16]. Conventional surrogates may fail when there is a strong periodic component in the original signal, as in our case. A few surrogate techniques for testing cyclic and pseudoperiodic time series have been proposed [18,7,22].

We have applied the pseudoperiodic surrogate (PPS) algorithm [18], whose surrogates do not suffer from stationarity and continuity problems. The PPS algorithm first constructs a vector delay embedding from the scalar time series in the phase space. Then, a random near neighbour of an initial point is selected and the procedure is iterated until a random time series with the same length as the original one has been generated. This method may be applied to test against the null hypothesis of a quasiperiodic orbit with uncorrelated noise, roughly preserving pseudoperiodic behavior. In our work, embedding dimension has been selected according to the Cao's method [1] and the embedding lag has been determined as the first minimum of the mutual information function [3], as this value minimizes the dependence degree between delayed samples.

For each subject, a set of 99 PPS surrogates is generated. Then, a complexity measure (see Section 3) is computed for each subject's surrogate and their mean and standard deviation values are attributed to that particular subject. Finally, the correlation between mean values and age is analyzed for surrogate means.

3 Complexity Measures

3.1 Approximate Entropy

The *ApEn* measure estimates the repeatability degree of short evolutive patterns throughout the complete data series [11]. It provides a non-negative finite index, where high values suggest high complexity, irregularity and unpredictability in the recorded signal [19,2]. *ApEn* can be applied to short time series, but it is sensitive to noise. Equal length records are required for different subjects.

Given a time series $x = \{x_1, \ldots x_n\}$, its approximate entropy $ApEn(m, r)$ is obtained as follows. In the first place, the $n - m + 1$ different m-dimension vectors $y(i) = [x_i, x_{i+1} \ldots x_{i+m-1}]$ are extracted. Let the distance between two vectors $y(i)$ and $y(j)$ be the greatest absolute value of the differences between their components:

$$d[y(i), y(j)] = \max_{0 \leq k \leq m-1} (|x_{i+k} - x_{j+k}|) \tag{1}$$

Then, we count the number $N_r^m(i)$ of m-dimension vectors $y(j)$ within a distance r of a given vector $y(i)$, and compute $C_r^m(i)$ as the ratio of $N_r^m(i)$ to the total number of m-dimension vectors in the time series:

$$C_r^m(i) = \frac{N_r^m(i)}{N - m + 1} \tag{2}$$

That is, $C_r^m(i)$ approximates to the probability of finding any m-dimension vector $y(j)$ similar to the vector $y(i)$ within a tolerance factor r. Then, the logarithmic average over i of the $C_r^m(i)$ probability is defined as:

$$C_r^m = \frac{\sum_{i=1}^{N-m+1} \ln C_r^m(i)}{N - m + 1} \tag{3}$$

The approximate entropy $ApEn(m,r)$ is then defined as $ApEn(m,r) = C_r^m - C_r^{m+1}$. The $ApEn(m,r)$ value is related to the probability that sequences that are similar for m samples remain similar for $m+1$ samples. A high value of $ApEn(m,r)$ means that we can not accurately predict the next sample from the knowledge of the previous m samples. In practice, $m = 2$ is a good value for short time series where high unpredictability is expected, and the tolerance factor r is selected as one or two times the standard deviation of the time series.

3.2 LZ Complexity

The Lempel-Ziv [8] complexity measure (LZ) assess the regularity and randomness in a symbol sequence. It compares the length of the original sequence (n) with the shorter length of the compressed sequence that can be obtained when repetitive subsequences are coded by means of a single index. A small value for the LZ complexity parameter γ indicates that highly repetitive non-random patterns are present in the sequence.

The basic procedure is as follows [9]. First, the signal is reduced to a sequence of n binary symbols. Then the sequence is parsed from left to right in order to construct a new compressed sequence by applying two basic operations, *copy* and *insert*. During this process, a set of words (or dictionary) that contains all the subsequences previously found in the already parsed segment of the original sequence is built. The *insert* action simply adds the present symbol to the compressed sequence. The *copy* action is applied when the present and preceding symbols form a pattern already coded in the dictionary. In this case, a dot is placed in the compressed sequence. When the parsing reaches the end of the given original sequence, a shorter compressed sequence has been generated. The length of this compressed sequence depends on the presence of repetitive subsequences, and it is smaller for highly redundant sequences.

At the end of this procedure, the Kolmogorov complexity $c(n)$ is computed as the length of the shortest compressed sequence that can be generated by the previously described procedure and it is associated to a kind of minimal length description. This value has to be normalized by $b(n) = n/log_2 n$, in order to enforce the Shannon entropy of the given sequence to be 1. The LZ complexity index is then defined as $\gamma = c(n)/b(n)$.

3.3 Instantaneous Phase Complexity

In this section we propose a novel approach to measure phase complexity in circadian rhythms. It is based on a simple idea: the application of the conventional complexity measures *ApEn* and *LZ* to the instantaneous phase of recorded signals. We consider that the presence of phase variability and phase shifts might be associated to a lack of circadian robustness. Complexity measures of instantaneous phase might be an appropriate indicator of cycle robustness. In this paper, instantaneous phase has been computed by means of the Hilbert transform and by means of a complex continuous wavelet transform (CWT).

Given a narrow band signal $x(t)$, its Hilbert transform is computed as [17]:

$$x_H(t) = \pi^{-1} PV \int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau} d\tau \tag{4}$$

where PV is the Cauchy Principal Value method for evaluating improper integrals. An analytic signal $\chi(t)$ can be obtained from x_H as $\chi(t) = x(t) + ix_H(t)$. The instantaneous phase is then defined as:

$$\varphi = \arctan \frac{x_H(t)}{x(t)} \tag{5}$$

In this way φ lies in the interval $(-\pi, \pi]$. φ is called the wrapped instantaneous phase and it has discontinuities. In order to obtain meaningful results when applying the *LZ* and *ApEn* complexity measures, the wrapped phase is required. Examples of wrapped phases are shown in Figures 1(a) and 1(b). φ depends on the magnitude of the DC continuous component, since the DC component is lost in $x_H(t)$, but not in $x(t)$ itself. In order to normalize, a zero-mean signal $x(t)$ is required (that is simply obtained by removing the average value from the signal).

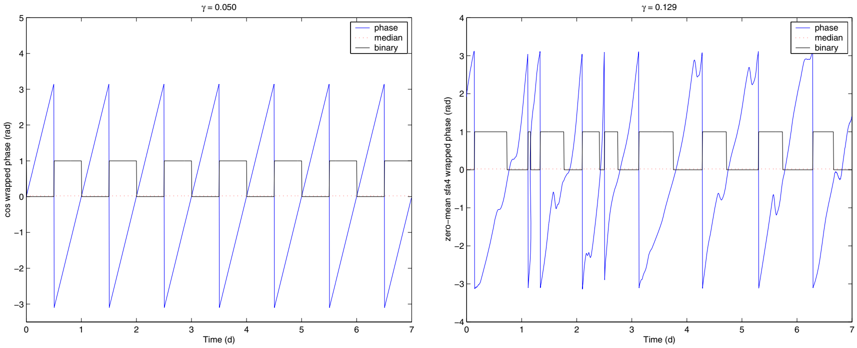
Additionally, we propose the application of the complex 1st order gaussian CWT at scale 43 as an alternate way to compute the instantaneous phase:

$$\varphi = \arctan \frac{Im[C(x)]}{Re[C(x)]} \tag{6}$$

Here x is an LP filtered signal and $C(x)$ is the CWT of the signal at the given scale. This method also provides values of φ in the interval $(-\pi, \pi]$.

We have studied three kind of phase complexity measures: the *LZ* complexity index γ of the Hilbert instantaneous phase (*LZHP*), the approximate entropy *ApEn* of the Hilbert instantaneous phase (*ApEnHP*), and the approximate entropy *ApEn* of the CWT instantaneous phase (*ApEnWP*).

On the one hand, for a pure sinusoidal function, unwrapped instantaneous phase is a perfectly linear decaying line. A lack of regularity in the signal cycles produces an unwrapped phase that is quasi-linear but exhibits irregularities and occasional phase shifts. On the other hand, the wrapped instantaneous phase of a pure sinusoidal signal has an approximate sawtooth shape perfectly regular (Figure 1(a)). However, a lack of regularity in the signal cycles produces an irregularly spaced sawtooth shape that is strongly affected by occasional small phase



(a) *LZHP* complexity (*cos*)

(b) *LZHP* complexity (*sfa4*)

Fig. 1. Examples of instantaneous phases and *LZHP* complexity

shifts (Figure 1(b)). In this case, the binary version for the *LZ* algorithm is a rectangular signal with varying pulse widths. The *LZ* complexity index measures the degree of disorder on this quasi-rectangular signal. The *ApEn* complexity index measures the degree of short-term unpredictability in phase evolution by comparing small fractions of the irregular sawtooth shaped instantaneous phase.

4 Results and Discussion

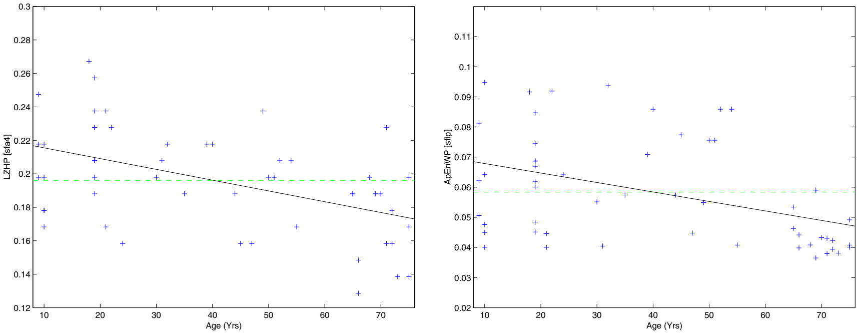
Table 1 shows the results achieved for instantaneous phase complexity measures *LZHP*, *ApEnHP* and *ApEnWP*. They have been applied to the *sfa4* and *sflp* signals, in order to obtain some insight about the dependence on the signal bandwidth. Results for PPS surrogate means (*PPS sfa4* and *PPS sflp*) are also shown. Surrogates have been preprocessed in the same way that the original signal. The parameter β_1 is the slope of the linear regression, r is the corresponding correlation coefficient and ρ is the Spearman correlation coefficient. Figures 2(a) and 2(b) show that *LZHP* and *ApEnWP* complexity indexes decrease with age.

The values $m = 2$ and $r = 0.2 \times std$ have been selected for the $ApEn(m, r)$ statistic (see Section 3.1). In the case of LZ complexity, the median is used as the threshold for binary conversion because of its robustness to outliers and normalization properties [9]. In the case of *ApEn* complexity of CWT instantaneous phase (*ApEnWP*), complex gaussian wavelets have been selected because of its symmetry and smoothness. The particular combination of order and resolution (1st order cgau1 at scale 43) has been selected because it corresponds to the frequency 1.01 c/d, quite close to the circadian normal frequency. CWT allows the use of a continuous scale, while in DWT only discrete 2^l scales are allowed.

In general, results show that phase complexity significantly decreases with age. Our results suggest that circadian phase irregularities and phase shifts are less frequent when ageing. *LZHP* and *ApEnHP* appear as the best behaved statistics for phase complexity. In fact, both indexes, when rounded, provide

Table 1. Analysis of *LZHP*, *ApEnHP* and *ApEnWP* correlation with age

	Signal	β_1	r	p	ρ	ps
LZHP	*sfa4*	$-0.644 \cdot 10^{-3}$	0.492	$0.182 \cdot 10^{-3}$	−0.465	$0.454 \cdot 10^{-3}$
LZHP	*PPS sfa4*	$-0.203 \cdot 10^{-3}$	0.278	0.044	−0.259	0.062
LZHP	*sflp*	$-0.738 \cdot 10^{-3}$	0.469	$0.399 \cdot 10^{-3}$	−0.448	$0.770 \cdot 10^{-3}$
LZHP	*PPS sflp*	$-0.269 \cdot 10^{-3}$	0.290	0.035	−0.261	0.059
ApEnHP	*sfa4*	$-1.016 \cdot 10^{-3}$	0.492	$0.182 \cdot 10^{-3}$	−0.473	$0.354 \cdot 10^{-3}$
ApEnHP	*PPS sfa4*	$-0.514 \cdot 10^{-3}$	0.344	0.012	−0.317	0.021
ApEnHP	*sflp*	$-0.795 \cdot 10^{-3}$	0.431	$1.270 \cdot 10^{-3}$	−0.381	$4.849 \cdot 10^{-3}$
ApEnHP	*PPS sflp*	$-0.546 \cdot 10^{-3}$	0.369	0.007	−0.355	0.009
ApEnWP	*sfa4*	$-0.288 \cdot 10^{-3}$	0.389	$3.982 \cdot 10^{-3}$	−0.440	$0.980 \cdot 10^{-3}$
ApEnWP	*PPS sfa4*	$-0.054 \cdot 10^{-3}$	0.363	0.008	−0.368	0.007
ApEnWP	*sflp*	$-0.315 \cdot 10^{-3}$	0.417	$1.880 \cdot 10^{-3}$	−0.463	$0.488 \cdot 10^{-3}$
ApEnWP	*PPS sflp*	$-0.045 \cdot 10^{-3}$	0.296	0.032	−0.301	0.029



(a) *LZHP* γ vs age (*sfa4*) (b) *ApEnWP* vs age (*sflp*)

Fig. 2. Correlation of some phase complexity measures with age

identical parameters for linear regression, although *ApEnHP* has a slightly better Spearman's rank correlation. However, *LZHP* exhibits the smallest sensitivity to the signal bandwidth and the largest differences with surrogates. In fact, *ApEnHP* fails the surrogate test for the *sflp* signal, but not for *sfa4*. Conversely, *ApEnWP* fails the surrogate test for the *sfa4* signal, but not for *sflp*. In summary, *LZHP* might be considered a more robust index.

Our analysis suggests that the signal bandwidth is somewhat critical for the *ApEn* based measures. PPS surrogates seem to be too similar to the original signal, perhaps because they retain some feature of the periodic orbit in the phase space that *ApEn* statistic can not discriminate solely based on linear regression. Further studies are needed to assess the dependence on bandwidth and different kinds of cyclic surrogates should be tested. It is worth to note that the mean value of the *LZHP* index for *sfa4* signals is $\gamma = 0.196 \pm 0.031$, while for a pure 1 c/d sinusoidal signal it is $\gamma = 0.050$. This seems to reveal a relatively high phase complexity level in healthy circadian temperature rhythms.

5 Conclusions

We have presented a study about complexity changes in human wrist temperature circadian rhythms through ageing. The main contributions in this work is the introduction of some measures of instantaneous phase complexity, based on Hilbert transform and CWT. *LZHP* phase complexity is decoupled from global signal complexity, thus providing independent information. Our results consistently show a decrease in phase complexity with age.

Previous works [19,2] have studied *ApEn* and DFA in circadian skin temperature records. In our case, we have analyzed larger records ($7d$ vs $30h$), with special emphasis in the signal bandwidth, surrogate tests and phase complexity.

Weinert [20] has signaled that the changes in circadian rhythms when ageing appear related to changes within the SCN central pacemaker. The total number of neurons in SCN does not change with age, so changes must be due either to changes in the molecular clock inside the cells or to alterations in the coupling mechanism between individual neurons. On the one hand, there is strong evidence showing that the molecular clockwork inside the cells remains unaffected with age. On the other hand, experimental work shows that the coupling mechanism between neurons deteriorates with age, and that both neuronal firing rates and neurotransmitters production in neurons involved in the circadian pacemaker also decreases. Weinert concludes that the ability of the aged SCN to transmit the circadian signal to peripheral systems may be diminished [20]. We are tempted to conjecture that in these circumstances, the simpler peripheral clocks could acquire prominence over the central pacemaker signaling, thus exhibiting decreased complexity and increased regularity.

References

1. Cao, L.: Practical method for determining the minimum embedding dimension of a scalar time series. Physica D: Nonlinear Phenomena 110(1-2), 43–50 (1997)
2. Cuesta, D., Varela, M., Miró, P., Galdós, P., Abásolo, D., Hornero, R., Aboy, M.: Predicting survival in critical patients by use of body temperature regularity measurement based on approximate entropy. Medical and Biological Engineering and Computing 45(7), 671–678 (2007)
3. Fraser, A.M., Swinney, H.L.: Independent coordinates for strange attractors from mutual information. Physical Review A 33(2), 1134–1140 (1986)
4. Gonze, D., Goldbeter, A.: Entrainment versus chaos in a model for a circadian oscillator driven by light-dark cycles. Journal of Statistical Physics 101(1), 649–663 (2000)
5. Hofman, M.A., Swaab, D.F.: Living by the clock: the circadian pacemaker in older people. Ageing Research Reviews 5(1), 33–51 (2006)
6. Huang, Y.L., Liu, R.Y., Wang, Q.S., Van Someren, E.J.W., Xu, H., Zhou, J.N.: Age-associated difference in circadian sleep-wake and rest-activity rhythms. Physiology & Behavior 76(4-5), 597–603 (2002)
7. Keylock, C.J.: A wavelet-based method for surrogate data generation. Physica D: Nonlinear Phenomena 225(2), 219–228 (2007)

8. Lempel, A., Ziv, J.: On the Complexity of Finite Sequences. IEEE Transactions on Information Theory 22(1), 75–81 (1976)
9. Nagarajan, R., Szczepanski, J., Wajnryb, E.: Interpreting non-random signatures in biomedical signals with Lempel-Ziv complexity. Physica D: Nonlinear Phenomena 237(3), 359–364 (2008)
10. Ortiz-Tudela, E., Martinez-Nicolas, A., Campos, M., Rol, M.A., Madrid, J.A.: A new integrated variable based on thermometry, actimetry and body position (TAP) to evaluate circadian system status in humans. PLoS Computational Biology 6(11) (2010)
11. Pincus, S.M.: Approximate entropy as a measure of system complexity. Proceedings of the National Academy of Sciences of the United States of America 88(6), 2297 (1991)
12. Refinetti, R.: Non-stationary time series and the robustness of circadian rhythms. Journal of Theoretical Biology 227(4), 571–581 (2004)
13. Refinetti, R., Cornélissen, G., Halberg, F.: Procedures for numerical analysis of circadian rhythms. Biological Rhythm Research 38(4), 275–325 (2007)
14. Sarabia, J.A., Mondejar, M.T., Marin, R., Campos, M., Rol, M.A., Madrid, J.A.: Caracterizacion del ritmo de temperatura periferica de la mueca, en sujetos monitorizados ambulatoriamente. In: Proceedings de la Tercera Reunion Nacional de la Sociedad Espa ola de Medicina Geriatrica (SEMEG), Oviedo, Spain (2008)
15. Sarabia, J.A., Rol, M.A., Mendiola, P., Madrid, J.A.: Circadian rhythm of wrist temperature in normal-living subjects: A candidate of new index of the circadian system. Physiology & Behavior 95(4), 570–580 (2008)
16. Schreiber, T., Schmitz, A.: Surrogate time series. Physica D: Nonlinear Phenomena 142(3-4), 346–382 (2000)
17. Shiogai, Y., Stefanovska, A., McClintock, P.V.E.: Nonlinear dynamics of cardiovascular ageing. Physics Reports 488(2-3), 51–110 (2010)
18. Small, M., Tse, C.K.: Applying the method of surrogate data to cyclic time series. Physica D: Nonlinear Phenomena 164(3-4), 187–201 (2002)
19. Varela, M., Jimenez, L., Fariña, R.: Complexity analysis of the temperature curve: new information from body temperature. European Journal of Applied Physiology 89, 230–237 (2003)
20. Weinert, D.: Circadian temperature variation and ageing. Ageing Research Reviews 9(1), 51–60 (2010)
21. Weinert, D., Waterhouse, J.: The circadian rhythm of core temperature: effects of physical activity and aging. Physiology & Behavior 90(2-3), 246–256 (2007)
22. Zhang, J., Luo, X., Small, M.: Detecting chaos in pseudoperiodic time series without embedding. Physical Review E 73(1), 16216 (2006)

Radial Basis Function Neural Network for Classification of Quantitative EEG in Patients with Advanced Chronic Renal Failure

Juan A. Barios[1], César Gonzalez[2], Bettina Benbunan[3],
Victor Fernández-Armayor[3], José L. Teruel[1], Milagros Fernández[1],
Antonio Pedrera[1], and José M. Gaztelu[1]

[1] Instituto de Investigación Ramón y Cajal (IRYCIS),
Hospital Universitario Ramón y Cajal,
Ctra. Colmenar km 9, Madrid 28034, Spain
jose.m.gaztelu@hrc.es
[2] Deimos Space SL, Cantoblanco, Madrid, 28760, Spain
[3] Unidad de Ciencias Neurológicas, Madrid 28040, Spain

Abstract. In this study we investigate the potential of applying the radial basis function (RBF) neural network architecture for the classification of patients with chronic renal failure (CRF) through quantitative parameters derived from EEG. To provide an objective EEG assessment of cerebral disturbances in CRF, we set up and tested a procedure of classification based on artificial neural networks (ANN) using RBF trained with quantitative parameters derived from EEG. A set sample was prepared based on EEG of 17 patients and 18 age-matched control subjects. Quantitative EEG (qEEG) found significant differences between groups. Accuracy of ANN-based classification in this set was 86.6%. Our results indicate that a classification system based on RBF neural networks may help in the automation of EEG analysis for diagnosis and prospective clinical evaluation of CRF patients.

Keywords: chronic renal failure, hemodialysis, qEEG, signal analysis, artificial neural network, classification, radial basis function.

1 Introduction

Patients with CRF frequently have impaired cognitive functions (such as attention and memory) and mood alterations interfering with their quality of life. They often also have some degree of anxiety and depression which may be associated with organic symptoms, limitations imposed by the CRF and hemodialysis, the unique alternative to renal transplantation. Optimization of dialysis protocol is so a desired objective for improvement of patients treatment and quality of life.

Bio-signals contain indicators of current disease or warnings about impending diseases, and are usually used for diagnosis and follow up in clinical environments, providing researchers with great amounts of data to process. Computer

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 411–418, 2011.

based analytical tools can be very useful for in-depth study and classification of data. Artificial neural networks can perform the necessary transformation and clustering operations automatically and simultaneously. Neural networks have recently become widely accessible for automatic processing of EEG, thanks to increased availability of computing facilities. Several studies have presented the performance of neural network systems when used for the detection and recognition of abnormal EEGs, related to spike detection in epilepsy [1].

EEG has a long tradition as a useful neurophysiological tool in the study of metabolic encephalopathies, because of non invasiveness, easy performance, wide availability of implementation in clinical settings, and low cost. Studies from our laboratory have shown that the EEG of patients with low grade hepatic encephalopathy have a marked slowdown, as well as significant changes detectable by linear and nonlinear analysis techniques [2–4]. In the encephalopathy associated with CRF some EEG findings have also been described, as slowing of background activity and increases in spectral power in slow frequency bands [5–7].

The aim of the present study was to investigate the potential of applying the RBF neural network architecture for prospective evaluation of cerebral function in CRF patients based on relevant characteristic features extracted from EEG.

2 Methods

The proposed approach for the classification of CRF patients involves preprocessing of the EEG signal, extraction of characteristic features and classification using articial neural networks (ANN) techniques. Using RBF based ANN techniques, the subjects were classified into two classes: normal (control subjects) and abnormal (renal disease present).

2.1 Patients

We selected 17 patients with CRF without dementia, 7 men and 10 women, aged between 29 and 82 years (mean 52.86, SD = 17) studied at the Nephrology Service of Ramn y Cajal Hospital in Madrid, and a control group of 18 age-matched patients with normal EEG and absence of metabolic illness. The main demographic, clinical and biochemical criteria of the patients are reported in table 1.

Table 1. Clinical description of groups

Parameter	Control	CRF
Cases (n)	18	17
Sex (M/F)	9/9	10/7
Age	57.2±17.6	56.0±15.1
Time in dialysis (months)		30.9±28.5
Glomerular filtration (ml/min)		12.84±3.44
Creatinine (mg/dl)		7.63±3.00

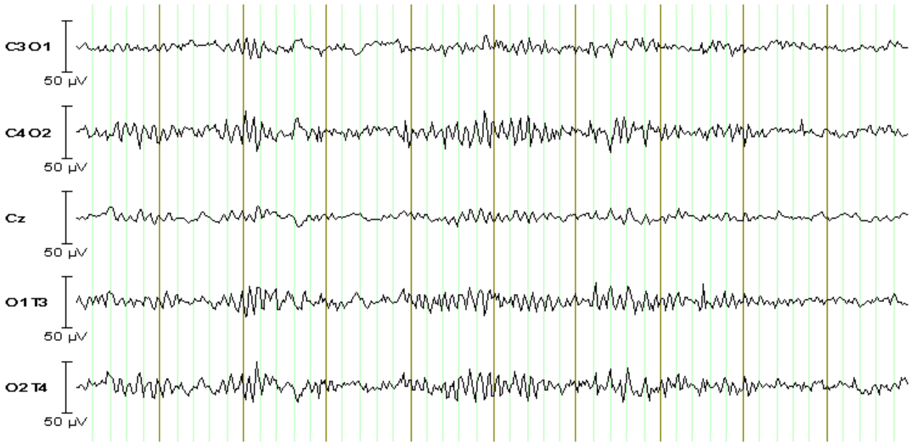


Fig. 1. Example of raw EEG (normal subject). Time marks, 1 second.

2.2 Recording and Visual Analysis of the EEG

EEG recordings were obtained before the beginning of the hemodialysis session in psycho-physical rest for 20 minutes, alternating eyes open and closed and activation is performed with 3 minutes of hyperventilation and intermittent light stimulation at 8, 10, 15 and 20 Hz. An example of normal EEG is shown in figure 1. To record the signal, a video-EEG equipment (Neuroworks Xltek version 5.4.0) of 32 channels was used. The EEG signal was recorded using surface electrodes, $AgCl$, once an impedance of less than $10\,K\Omega$ was reached. Using the 10/20 International System for electrode placement, scalp electrodes were placed at Fp1, F3, C3, P3, O1, F7, T3, T5, Fz, Cz, Pz, Fp2, F4, C4, P4, O2, F8, T4, T6 and A1 as reference. Recording was performed with a longitudinal mounting: Fp1-F3, F3-C3, C3-P3, P3-O1, Fp2-F4, F4-C4, C4-P4, P4-O2, Fp1-F7, F7-T3, T3-T5, T5 -O1, Fp2-F8, F8-T4, T4-T6, T6-O2, Fz-Cz, Cz-Pz, and other reference (each against a common reference electrode used: A1). The sampling frequency was 500 Hz, high pass filters set at 0.5 Hz and low pass at 100 Hz, with AC filter (notch filter) at 50 Hz. Records were stored on computer disk for off-line analysis. The initial visual inspection was performed with a scanning screen of 15 seconds (30 mm/second) and with the scale extended to 10 μV/mm.

2.3 Quantitative Analysis of the EEG

Signals were exported to MATLAB environment. After semiautomatic elimination of artifacts, with visual confirmation, quantitative analysis of EEG was performed in each of the derivations using different numerical techniques. Power spectrum was calculated using Welch averaged periodogram, with a window of 4 seconds, 50% overlap and a Hanning window. Blocks containing artifacts were skipped when averaging the periodograms. Relative power was evaluated in the

Table 2. qEEG extracted parameters used for training of ANN

Parameter	Frequency range
MDF	1-26 Hz
MDF	4-12 Hz
Spectral entropy	
Sample entropy	
δ relative power	0.54 Hz
low δ relative power	0.52 Hz
high δ relative power	24 Hz
θ relative power	4-8 Hz
α relative power	8-12 Hz
β relative power	15-25 Hz
γ relative power	25-40 Hz

following frequency bands: δ (0.75-4 Hz), θ (4-8 Hz), α (8-12 Hz), β (15-25 Hz), γ (25-40 Hz). Linear parameters derived from power spectral density (mean dominant frequency (MDF) and spectral entropy) and non-linear (sample entropy) were also calculated. MDF was calculated as the mean of frequency weighted by power spectral density. Spectral entropy was calculated as the Shannon entropy

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x))$$

of spectral density. Sample entropy, introduced in [9], measures the probability that two subsequences similar in their first m points (with distance $< r$) are still similar for $m + 1$. It was calculated using free downloadable subroutine $sampen(N, m, r)$: N is the number of points of the sample (length segment=1 second, 500 points), $m = 3$, $r = 0.2$ [9].

In figure 2, we show the symmetric scatter plot matrix containing all pairwise scatter plots for the calculated qEEG parameters and the corresponding values of the pearson coefficient. Each scatter plot was generated by pooling feature values for all subjects. Perfect agreement between features i and j would show up as a straight line in row i, column j. If there were no correlation between the two features, the corresponding scatter plot would display a “cloud” of points. MDF, spectral entropy and β power seem to be highly correlated. α power has a high negative correlation with δ power. On the other hand, sample entropy, appears to be less correlated to any other parameter. So, the chosen subset of parameters seems to explore orthogonal aspects of signal.

2.4 Artificial Neural Network Analysis

After confirming the absence of significant differences between left and right side, parameters obtained for each of the 19 EEG derivations were pooled in five groups grouped by antero-posterior situation of electrodes: frontal (F3-Fz-F4), central (C3-Cz-C4), parietal (P3-Pz-P4), temporal (T3-T4) and occipital (O1-Oz-O2). 11 characteristic parameters were calculated for the recording in each

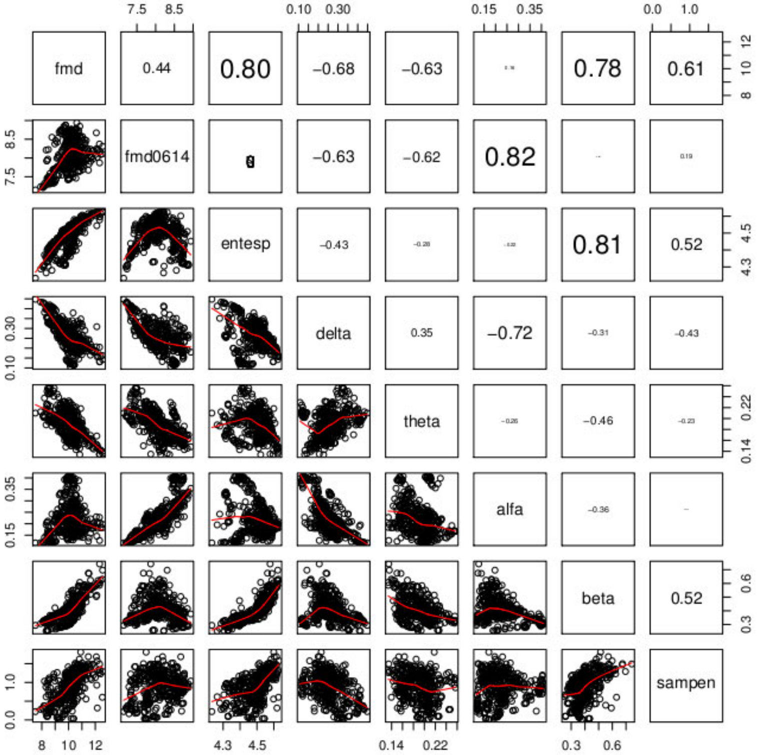


Fig. 2. Scatter plot matrix and corresponding values of the Pearson coefficient between chosen parameters derived from qEEG

EEG derivation. The parameters are listed in table 2. So, 55 parameters were obtained for each subject.

Using these characteristic features as the input set, a neural network classifier using RBF is implemented as a custom made MATLAB script.

The net input to the radial basis transfer function is the vector distance between its weight vector, w, and the input vector, p, multiplied by the bias, b. The RBF has a maximum of one when its input is zero. As the distance between w and p decreases, the output increases. Thus, a radial basis neuron acts as a detector, which produces one whenever the input, p, is identical to its weight vector, w. A probabilistic neural network, a variant of radial basis network, was used for the classifications. When an input is presented, the first layer computes distances from the input vector to the training input vectors and produces a vector whose element indicates how close the input is to a training input. The second layer sums these contributions for each class of inputs to produce net output vector probabilities. Finally, a complete transfer function on the output of the second layer picks the maximum of these probabilities and produces a one for that class and a zero for the other classes.

Table 3. qEEG results in experimental and control group

Parameter	Control	CRF	p-value
MDF 1-25 Hz	10.65 ±0.71	10.11±0.86	<0.001
MDF 6-14 Hz	8.20±0.37	8.09±0.42	0.062
Spectral Entropy	4.52 ±0.065	4.48±0.081	0.0022
δ relative power [0.75-4 Hz]	0.234±0.062	0.274±0.081	0.00102
δ_1 relative power [0.75-2 Hz]	0.136±0.042	0.162±0.056	0.00112
δ_2 relative power [2-4 Hz]	0.0985±0.023	0.112±0.027	0.00233
θ relative power [4-8 Hz]	0.145±0.281	0.156±0.0242	0.0101
α relative power [8-12 Hz]	0.197±0.069	0.191±0.072	n.s.
β relative power [15-25 Hz]	0.376±0.052	0.340±0.055	<0.001
γ relative power [25-40Hz]	0.097±0.031	0.101±0.032	n.s.
Sample Entropy	0.796±0.36	0.916±0.35	0.0414

The last layer is given by the probablitity of belonging to experimental or control groups. Therefore, different epochs of the EEG trace were classified by the network in different categories. 20 examples were automatically presented to the ANN (10 experimental and 10 controls) until these were correctly classified between experimental and control groups, and rest of cases were presented to test the ANN.

2.5 Statistics

The statistical contrasts (table 3) were performed in each case using the t-Student or Mann-Whitney test, depending on whether or not compliance with the assumption of normality in the data. Correlations were assessed using Pearson correlation. The α significance level was set to 0.05, although values under 0.1 will be reflected. The statistics were calculated using open-source program R [10].

3 Results

3.1 EEG Results

Visual analysis: Visual analysis of EEG, reported by an expert (physician specialist in clinical neurophysiology), reported minor alterations: 13 traces were normal, 3 of low voltage, 1 with presence of δ bursts during hyperventilation (normal variant).

qEEG: We found significant differences in qEEG between control and experimental groups in almost all studied parameters. The experimental group presented reduced MDF, MDF$_{6-14Hz}$, spectral entropy and β power, and increased δ and θ power and sample entropy (see table 3).

3.2 ANN Results

In the supervised classifications, available EEGs were distributed in two sets, training and testing sets. The size of the training dataset was 20 EEGs (10 controls and 10 uremic patients). In order to define the most significant parameters, different combinations of inputs have been used (from one unique band to all available data) for the training of the ANN. The rest of the EEG records were used for testing the network (7 uremic patients and 8 controls).

The classification results were not satisfactory when only one frequency band was selected for input; in those cases the classification accuracy was never higher than 60%. Using the whole of 55 input parameters, the accuracy was 80% (2 errors in uremic patients and only 1 error in controls). The highest classification accuracy was obtained when eliminating the MDF parameters. In that case, the ANN correctly clasified 86% of the cases (2 errors in uremic and no errors in controls).

These results show that no abnormalities can be found using just one frequency band of the EEG or any isolated parameter. The need of combining bands and parameters demonstrates that there are some complex correlations between bands in uremic patients, and that those patterns can be detected by the ANN. As final conclusions we can state that brain abnormalities in uremic patients are complex and should not be studied using just one parameter, and that the pattern of manifestations is not always the same in different patients.

4 Conclusions

In the present study, an automatic system for the classification of CRF patients employing ANN techniques was developed and implemented. The classification was performed using features extracted from EEGs. Emphasis was placed on selection of the characteristic features and for the accurate extraction of these features.

From the results, it can be seen that the RBF network provided an excellent performance for the studied application, classifying accurately a 100% of controls and a 86.6% of CRF patients. The performance of the system can be further enhanced by training with a larger number of training inputs, which would increase the network ability to classify unknown signals. The system performance can be also improved by considering other features of EEG that were not included in the current system. Thus, the present study shows the feasibility of the application of RBF based ANN for the classification of CRF patients.

References

1. Gabor, A.J., Seyal, M.: Automated interictal EEG spike detection using artificial neural networks. Electroencephalography and Clinical Neurophysiology 83(5), 271–280 (1992), ISSN: 0013-4694

2. Gaztelu, J.M., Martino, M.E., Fernández-Lorente, J., Romero-Vives, M., de Vicente, E., Bárcena, R.: Sleep EEG spectral characteristics disclose subclinical hepatic encephalopathy. In: Society for Neuroscience XXXI Annual Meeting. San Diego, California, November 10-15 (2001)
3. Martino, M.E.: Análisis cuantitativo del electroencefalograma en pacientes cirróticos. Implicaciones diagnósticas y terapéuticas. PhD Doctoral Thesis, UAM, Madrid (2003)
4. Barios, J.: Caracterización de la actividad oscilatoria cerebral durante el sueño lento. Alteraciones en la encefalopata hepática mínima y modicaciones con el trasplante hepático. PhD Doctoral Thesis, UAM, Madrid (2008)
5. Bourne, J.R., Ward, J.W., Teschan, P.E., Musso, M., Johnston, J.R., Ginn, H.E.: Quantitative assessment of the electroencephalogram in renal disease. Electroencephalography and Clinical Neurophysiology 39, 377–388 (1975)
6. Balzar, E., Saletu, B., Khoss, A.: Quantitative EEG: investigation in children with end stage renal disease before and after haemodialysis. Clinical EEG (electroencephalography) 17, 195–202 (1986)
7. Röhl, J., Harms, L., Pommer, W.: Quantitative EEG findings in patients with chronic renal failure. European Journal of Medical Research 12, 173–178 (2007)
8. Pellegrini, A., Ubiali, E., Orsato, R., Schiff, S., Gatta, A., Castellaro, A., Casagrande, A., Amodio, P.: Electroencephalographic staging of hepatic encephalopathy by an artificial neural network and an expert system. Neurophysiologie Clinique = Clinical Neurophysiology 35, 162–167 (2005)
9. Richman, J.S., Moorman, J.R.: Physiological time-series analysis using approximate entropy and sample entropy. American Journal of Physiology. Heart and Circulatory Physiology 278, H2039–H2049 (2000)
10. R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2008), http://www.R-project.org, ISBN 3-900051-07-0

Bayesian Network-Based Model for the Diagnosis of Deterioration of Semantic Content Compatible with Alzheimer's Disease

José María Guerrero Triviño, Rafael Martínez-Tomás
and Herminia Peraita Adrados

josemaria.guerrero@cpiia.org

Abstract. Alzheimer's Disease (AD) has become a serious public health problem that affects both the patient and his family and social environment, not to mention the high economic cost for families and public administrations. The early detection of AD has become one of the principal focuses of research, and its diagnosis is fundamental when the disease is incipient or even prodromic, because it is at these stages when treatments are more effective. There are numerous research studies to characterise the disease in these stages, and we have used the specific research carried out by Drs. Herminia Peraita and Lina Grasso. The application of Artificial Intelligence techniques, such as Bayesian Networks and Influence Diagrams, may provide a very valuable contribution both to the very research and the application of results. This article justifies using Bayesian Networks and Influence Diagrams to solve this type of problems and because of their great contribution to this application field. The modelling techniques used for constructing the Bayesian Network are mentioned in this article, and a mechanism for automatic learning of the model parameters is established.

Keywords: Bayesian Network, Influence Diagram, Corpus of Oral Definitions, Naive Bayes, Alzheimer's Disease, Cognitive Deterioration.

1 Introduction

As in other fields in the real world, medical diagnosis is not always 100% accurate. In the specific case of Dementia and especially Alzheimer's Disease (AD), its diagnosis is sometimes an extremely difficult task, especially when it is incipient and intensity is only slight[15]. These are the main reasons for using Soft Computing techniques (techniques that enable us to work with incomplete, inexact and uncertain information) to solve this type of problems. Bayesian Networks, in particular, provide a probabilistic model that makes it possible to define the causal relations between the variables explicitly. This causality is assigned a relation force, which is logically determined by the degree of correlation or causality between the variables. Bayesian Networks are extremely useful in response to new cases,and there are Automatic-Learning techniques for both qualitative and quantitative models. These Automatic Learning techniques can enable us to

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 419–430, 2011.

discover new relations between the variables or new conditional probabilities, as new cases appear. [10,2]

It has been demonstrated that there is a semantic deterioration in the ability to differentiate between living creatures (biological entities) and non-living creatures (non-biological entities) in people suffering certain neurodegenerative pathologies (Alzheimer, Semantic Dementia, Dementia with Lewy bodies, etc.), traumatic pathologies (cranial traumatism), and infectious pathologies (herpes encephalitis). Semantic categories are derived from classifications that are carried out in the world around us and that treat essentially different objects as the same. Thanks to the fact that our semantic memory is organised according to these categories, we can perform a series of important cognitive functions, such as inferring, establishing relations between examples, attributing properties to objects that we do not know, reasoning, all of which is based on a cognitive economy principle. People who suffer specific category deficits execute tasks affecting totally or partially the category domain knowledge of living creatures worse, whereas the object or artefact domain –non-living creatures- is totally or almost totally conserved. There are also a small number of cases where the pattern is the reverse; there is more deterioration in the object or artefact domain, whereas the living creature domain is largely preserved [13].

Thus, the Bayesian Network model constructed in this article aims to diagnose whether the patient suffers cognitive deterioration compatible with AD. This diagnosis is done from a corpus of oral definitions as a methodological tool that has been shown to be very useful to study pathologies in relation to the semantic deterioration of this disease [12]. It is a causal model based on literal definitions of certain semantic categories –of the basic level of categorisation—both of living creatures (dog, pine and apple), and non-living creatures (chair, car, trousers). Patients do some tests where they have to define basic objects. When a patient suffers AD, he suffers from serious cognitive deterioration. The attributes, features or characteristics generated by each patient's definitions are analysed. The underlying logic for analysing the suggested features is in accordance with a model described some time ago in Peraita, Elosúa and Linares (1992) and in keeping with other current works ([3,8,9]).

In the causal model we represent that AD causes a conceptual-semantic-lexical deficit and therefore the Bayesian Network will be able to infer the probability of suffering AD from the degree of conceptual-semantic-lexical deficit. This inference or abductive reasoning starts from some symptoms and searches for the causes that best explain the symptoms. In other words, we start from conceptual-semantic-lexical deterioration and search for the probability that this deterioration is the explanation for suffering AD. Some risk and protection factors, such as educational level, age and sex, will also be taken into account in the Bayesian Network.

Numerous works are currently being done in the field of Bayesian Networks like Early Diagnosis of Alzheimer's Disease. Works on Explanation in Bayesian Networks by [7][1] and other works of interest on Dynamic Bayesian Networks and Learning in Dynamic Bayesian Networks [5] should be highlighted.

Table 1. Abstract of Database of Instances (Linguistic Corpus)

PK	category	taxonomic	types	parts	functional	evaluative	place	conduct	cause	procedural	life cycle	other
1	car	0	1	5	5	0	0	0	0	0	0	3
1	apple	1	2	0	6	0	0	0	1	0	0	0
1	trousers	0	8	0	2	0	0	0	0	0	0	0
1	dog	0	6	0	1	4	0	0	0	0	0	1
1	pine	0	0	0	5	0	0	0	0	0	0	0
1	chair	0	4	0	2	0	3	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...

This work starts from a set of data obtained from the works by [12][14], where semantic category definition features of the linguistic Corpus of healthy subjects and subjects with Alzheimer's Disease are analysed. In other words, we start with a base of instances (table 1 shows an abstract) where the patients' definitions are classified into 11 basic conceptual blocks: taxonomical, types, parts, functional, evaluative, place/habitat, behaviour, causes/generates, procedural, life cycle and others. In other words, each patient's definition is analysed and the number of attributes that he produces is classified into these 11 basic conceptual blocks, with two differential semantic categories, living creatures and artefacts. The following table shows an abstract of the linguistic corpus (database of instances) that this research project has started from.

The research work is in its initial stage. Therefore, this article only describes the model for the diagnosis and the decisions that have been taken for its design. We think that in the very near future we can obtain promising results, especially when we have a wider database of instances, which will enable us to do sufficiently reliable and assessable experiments.

2 Justification for the Technique

Bayesian Networks and influence diagrams offer a number of advantages that make them attractive for use in this application field. The advantages can be summarised in [11,6]:

- Bayesian Networks are Soft Computing techniques and they have to be used because there is currently no deterministic method for diagnosing Alzheimer.
- Bayesian Networks are based on compact graphs and are intuitive of a causal relation between entities of a problem in a specific domain. Other techniques such as Neuronal Networks or Bioinspired Techniques are based on graph representations that are difficult for experts in the field to read.
- Inference is based on the theory of calculating probabilities and the decision theory. It therefore provides a coherent mathematical method to derive conclusions in uncertainty, where multiple sources of information are involved in complex interaction patterns. Other soft computing techniques are also based on complex mathematical models, where the inference explanation is very complex.

- Influence diagrams make it possible to take decisions in a normative way to establish the most appropriate action policy: complementary explorations, pharmacological and cognitive treatments, even for cases where it is not so obvious and the doctor's clinical judgement is unable to find the best solution,
- Decision analysis can explicitly and systematically combine different experts' opinions and experimental data, such as data from studies published in medical literature. [4]
- There are a number of analyses applicable to Bayesian Networks and Influence Diagrams, which provide added value to the technique, like for example, Evidence Conflict Analysis, Sensitivity Analysis (Evidence and Parameters), Value Analysis of the source of information.

There are Bayesian Network Frameworks that perform all or some of these analyses and they can be extremely useful for the problem that we are addressing, because they can also continually validate the model and facilitate the discovery of new relations, analyse new risk factors, new treatments, etc.

3 Proposal for the Diagnosis

As indicated earlier, the Bayesian Network consists of quantitative and qualitative models. For problem modelling four types of variables are used:

- Context information variables or risk factors. It is the information that is present before the problem occurs and that has a causal effect on the problem. In this group of variables we have: age, sex and educational level.
- Information variables representing the symptoms. Variables representing whether the patient has a conceptual-semantic-lexical deficit. These variables analyse the features or attributes contained in a number of semantic category definitions of living or non-living creatures, and other specific ones for each of these categories. The common ones are: taxonomical, functional, part-all, evaluative, place/habitat, types, and the uncommon ones: procedure, behavioural activity, cause/generation and origin. Attribute taxonomy, which acts as a schema or theoretical and methodological evaluation framework for this test, the same as in the second test, can be seen in detail in Peraita, Elosúa and Linares (1992). These variables represent attribute production in each semantic category of each of the objects used (apple, dog, pine, car, chair and trousers).
- Intermediate variables. They are variables that cannot be directly observed whose a posteriori probabilities are not of immediate interest, but they play an important role in achieving correct dependence and conditional independence of the properties and therefore efficient inference. Intermediate variables represent the semantic categories of living creatures and non-living creatures, the semantic content deficit of the different categories, etc. In this type of variables we have: Cognitive Deterioration Living and Non-Living Creatures on the one hand, and Cognitive Deterioration Apple, Dog, Pine, Chair and Trousers, on the other.

- Variables of interest or assumption. We calculate their a posteriori probability from the findings. These variables are: Suffers Cognitive Deterioration (Dementia) and Suffers Alzheimer's Disease.

Discrete Bayesian Network Modelling

Alzheimer and Cognitive deterioration variables of interest have some risk factors represented by the variables Educational Level, Age and Sex. It is worth highlighting that they are risk factors and not causes of the disease, for that reason canonical models (OR/MAX) cannot be used. Furthermore, there is a causal link between "Alzheimer" and "Cognitive Deterioration". According to scientific literature and epidemiological studies, the most common cause of dementia in the European Union is Alzheimer (around 50-70% of cases), other causes of dementia are: multiple cardiac arrest (around 30% of cases), Pick's disease, Lewy bodies and others.

Intermediate variables representing conceptual-semantic-lexical deterioration in the category domain are automatically treated, analysing and interpreting the patient's definitions, distributed in the 11 basic conceptual blocks considered as conceptual components underlying every organisation and representation of object categories (taxonomical, types, parts, functional, evaluative, place/habitat, behaviour, causes/generates, procedural, life cycle and others). Each of these blocks has an identifying lexical label, as indicated in the work by Dr. Herminia Peraita [12,14].

If a patient suffers AD in prodromic or incipient stage, there is a differential deterioration between the semantic categories Living Creatures and Non-Living Creatures. According to Dr. Herminia Peraita's study[12,14], Alzheimer's disease produces cognitive deterioration in Living Creatures before Non-Living Creatures or artefacts. Since AD patients usually have greater damage in the temporal limb areas in the early stages of the disease, they could show selective deterioration for living creatures. As the disease progresses, the damage becomes so omnipresent that mistakes occur in both domains with the same frequency. Therefore, there is a causal relation that qualitative and quantitative models can take into account in this study.

Another important factor that is modelled in the Bayesian Network is the bidirectional correlations between intermediate variables and variables of interest. For example, when a patient produces few attributes in the semantic category of Living Creatures, the probability increases of producing few attributes in the semantic category of Non-Living Creatures when the disease is advanced. Similarly, when the disease is incipient and is Dementia caused by AD, the production of few attributes in the semantic category of Living Creatures increases the probability of producing a greater number of attributes in the semantic category of Non-Living Creatures. In other words, there is a negative causal relation between the two semantic categories when the disease is incipient and a positive correlation when the disease is advanced. We should remember that Bayesian networks do not include cycles. With the intermediate variables it will therefore be possible to represent these circumstances. In fig.1 and fig.2, the modelling techniques used to eliminate the cycles of de bayesian red can be observed.

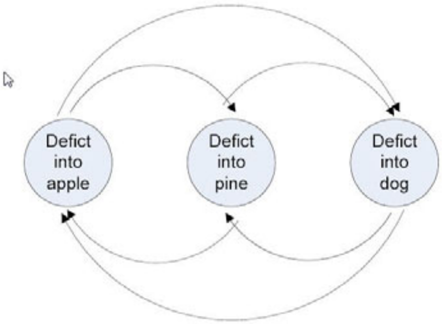


Fig. 1. Conditional dependencies between semantic categories. Graph with cycles.

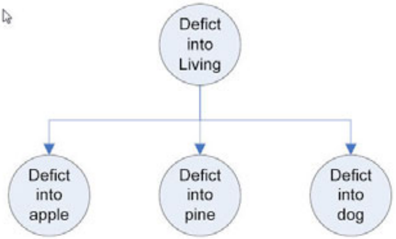


Fig. 2. Conditional dependencies between semantic categories. Graph without cycles.

The fig.3 represents the qualitative modelling of the Bayesian Network. The Construction of the qualitative model is one of the most complex tasks in the modelling of the Bayesian Network. In order to be able to generate the quantitative model automatically, it is necessary to have a large number of representative cases of the population. In many instances it is necessary to have epidemiological studies, and this task is not trivial at all. There are epidemiological studies that provide the distributions of combined probabilities thus: P(DEMENTIA, SEX), P(DEMENTIA, AGE), P(DEMENTIA, EDUCATIONAL LEVEL), P(ALZHEIMER, SEX), P(ALZHEIMER, AGE), P(ALZHEIMER, EDUCATIONAL LEVEL). In this instance it is very useful to use the Naive Bayes simplifier to calculate the conditional probability tables. The Naive Bayesian Method or Naive Bayes starts from the assumption that the diagnosis is exclusive (there cannot be two diagnoses at the same time) and exhaustive (there are no other diagnoses possible). In our model two conditions are fulfilled because we only aim to diagnose cognitive deterioration compatible with Alzheimer at three levels or degrees. With these premises we can use the Naive Bayes simplifier and it is therefore possible to calculate all the conditional probabilities that we need for the model, as can be shown below:

$$P(Alzheimer \mid sex,\ age,\ educational\ level) = a * P(Alzheimer) * P(sex \mid Alzheimer) * P(age \mid Alzheimer) * P(Educational\ Level \mid Alzheimer)$$

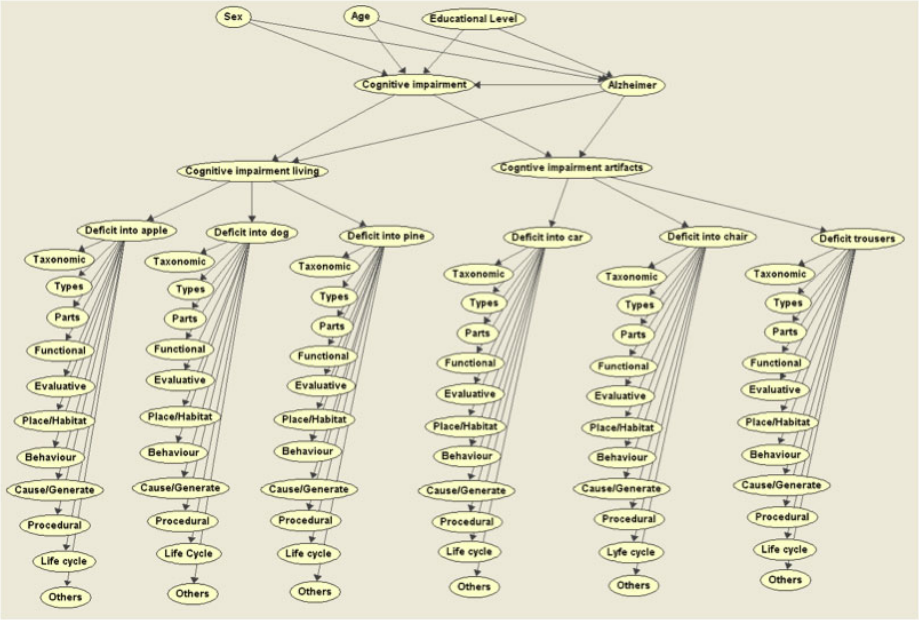


Fig. 3. Discrete Bayesian Network Modelling for the Diagnosis of Deterioration of Semantic Content

Where:

- α is a normalising constant.
- $P\left(sex \mid Alzheimer\right) = \frac{P(sex,Alzheimer)}{P(Alzheimer)}$
- $P\left(age \mid Alzheimer\right) = \frac{P(age,Alzheimer)}{P(Alzheimer)}$
- $P\left(EducationalLevel \mid Alzheimer\right) = \frac{P(EducationalLevel,,Alzheimer)}{P(Alzheimer)}$

For Cognitive Deterioration the same method can be used. Furthermore, an automatic learning algorithm can also be used for the intermediate variables and variables of interest. The total attribute production for each semantic category of each of the objects only has to be added. Then the K-Means algorithm is applied (using WEKA, fig) to create a specific number of data clusters, using the Euclidean distance for this. The K-Means algorithm is based on the patient producing attributes to determine the centroids of each cluster. Each cluster represents each of the states of the variable. Once the centroids of the different clusters have been defined, i.e. the different levels of cognitive deterioration, the database is analysed to calculate the conditional probabilities using the following formulation:

$$P\left(DCx_1 \mid DCSVx_2, FAx_2\right) = \frac{N(DCx_1,DCSVx_2,EAx_3)}{N(DCx_2,EAx_3)}$$

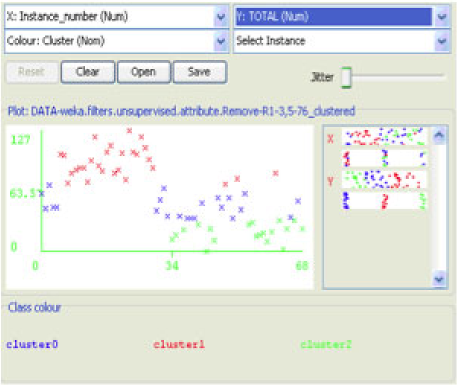
where

$CD \Rightarrow CognitiveDeterioration \Rightarrow \forall x_1 \in \{absent, slight, moderate, serious\}$

$CDLC \Rightarrow CognitiveDeteriorationLivingCreatures \Rightarrow \forall x_2 \in \{absent, slight, moderate, serious\}$

$AD \Rightarrow Alzheimer'sDisease \Rightarrow \forall x_3 \in \{absent, present\}$

N=> Is a function that counts the number of records in the database of instances that fulfil the query criteria.



X: Instance_number (Num)

Y: TOTAL (Num)

Colour: Cluster (Nom)

Select Instance

Plot: DATA-weka.filters.unsupervised.attribute.Remove-R1-3,5-76_clustered

Class colour

cluster0 cluster1 cluster2

Fig. 4. Clusters for cognitive deterioration generated with WEKA

Once the clusters for Cognitive Deterioration and Cognitive Deterioration in Living Creatures and Non-Living Creatures have been defined, each instance of the database has to be categorised according to the number of attributes that the patient has produced for each object. We have to make the symptom variables discrete, and for this we use the K-means algorithm, with as many clusters and states that we want the variables to have. Once the symptom variables are discrete, the database is analysed to obtain the conditional probabilities thus:

$$P\left(CATx_1 \mid DCOx_2\right) = \frac{N(CATx_1, DCox_2)}{N(DCOx_2)}$$

where

$$CAT \Rightarrow SemanticCategories \left\{ \begin{array}{c} Taxonomical \\ Types \\ Parts \\ Functional \\ Evaluative \\ Place/Habitat \\ Behaviour \\ Causes \\ Procedural \\ LyfeCycle \\ Others \end{array} \right\} \Rightarrow \forall x_1 \in$$
$$\{absent, slight, moderate, serious\}$$

$$DC \Rightarrow SemanticDeficit \left\{ \begin{array}{c} Apple \\ Dog \\ Pine \\ Car \\ Chair \\ Trousers \end{array} \right\} \Rightarrow \forall x_1 \in$$
$$\{absent, slight, moderate, serious\}$$

N=> Is a function that counts the number of records in the database of instances that fulfil the query criteria.

4 Influence Diagram

Influence Diagrams can be used as a Practical Clinical Guide to diagnose Alzheimer and to determine in a normative way which is the most appropriate action policy to take even for cases where it is not as evident and the doctor's clinical judgement cannot provide a better solution. They could also contribute to the application of more recent research studies, even explicitly and systematically combining different experts' opinions and the experimental data that are obtained. It should be borne in mind that in all the research process computer tools are being updated and upgraded and they are easily extensible For that reason, we have wanted to refer to the Influence Diagrams as a fundamental component for this research.

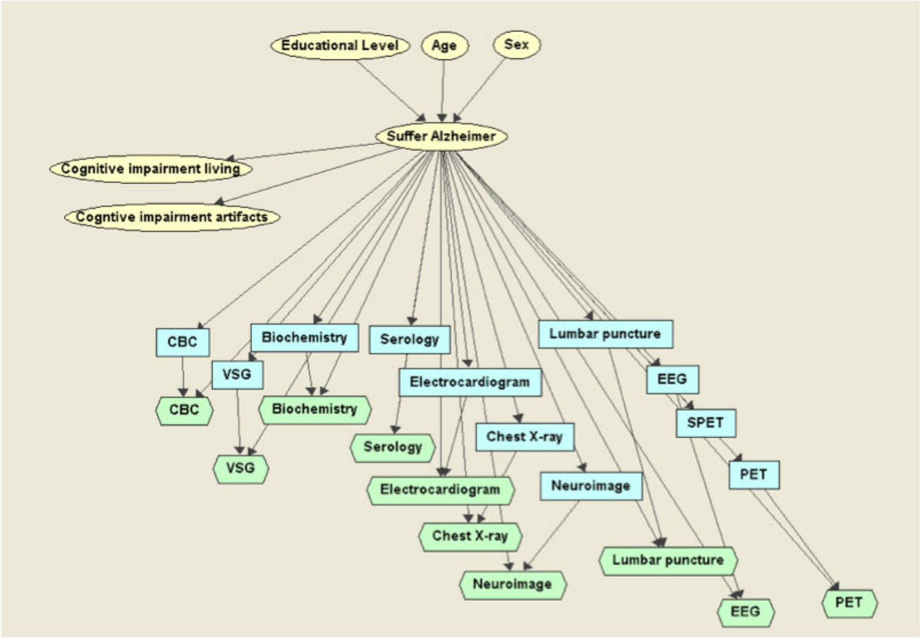


Fig. 5. Decision diagram for complementary explorations

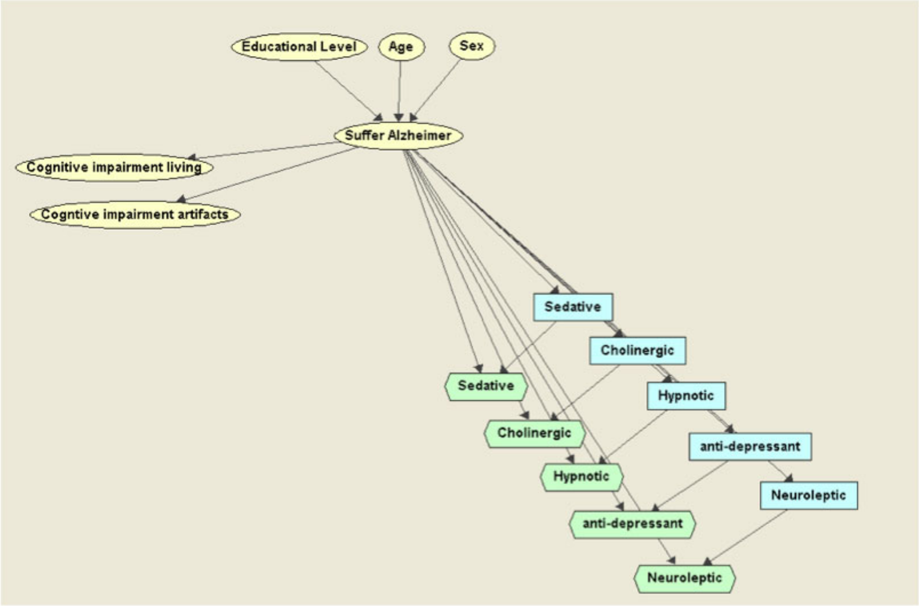


Fig. 6. Influence Diagram. Pharmacological treatments.

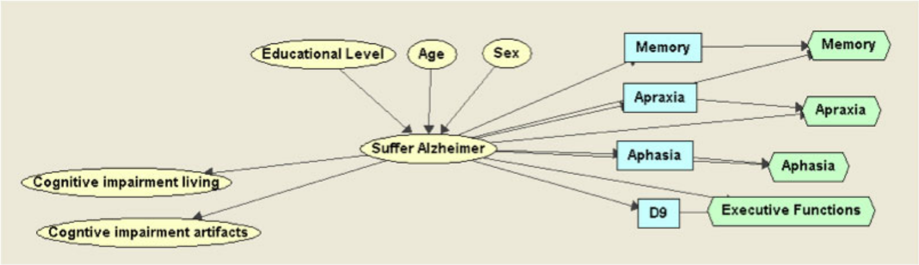


Fig. 7. Influence Diagram. Cognitive therapies.

This article proposes three influence diagrams: one (fig.5) to help determine whether complementary explorations must be done, another (fig.6) to help decide the most appropriate pharmacological treatment, and finally (fig.7), a decision diagram for cognitive treatment.

In the first Influence Diagram the following complementary explorations have been taken into account: CBC, VSG, Biochemical, Electrolytes, Hepatic function, Thyroid Function (T3, T4 and TSH), Vitamin (B12 and folic acid), Serology (Syphilis), Electrocardiogram, Neuroimage (CAT or Nuclear magnetic resonance), Lumbar Puncture, Electroencephalogram, SPECT (simple photon emission computed tomography) and/or PET (positron-emission tomography).

We have also constructed an Influence Diagram to analyse the maximum usefulness expected from each of the possible pharmacological treatments.

The last Influence Diagram evaluates the maximum usefulness expected in the application of cognitive therapies, i.e. psychological therapies based on the fundamentals of cognitive psychology.

5 Conclusions

This research work is in its initial stage, for that reason we have focused exclusively on the model and the decisions taken to design it. We believe that in the very near future we can obtain promising results, especially when we have a wider database of instances and more random instances in order to construct a realer quantitative model, which will enable us to do sufficiently reliable and assessable experiments. We are also convinced that Bayesian Networks can provide researchers with a powerful tool with great analytical capacity. Bayesian Networks and in particular the model that we present in this research work can include new variables, be they risk factors, symptoms or intermediate variables, and it is possible to analyse mathematically the impact that these variables can have on the diagnosis. This analytical capacity along with experts' epidemiological studies or subjective expert assessments could enable us to characterise the disease in its very initial stages. Furthermore, Influence Diagrams can establish action policies in a normative way, making it possible to apply and extend the most recent research studies in the Diagnosis of Alzheimer. Influence Diagrams could have great potential when applying and extending the research studies that are obtained.It should be highlighted that this research project opens a large multidisciplinary research field.

Acknowledgements

The authors are grateful to the CiCYT for financial aid on project TIN-2010-20845-C03-02.

References

1. Arias-Calleja, M.: Carmen: una herramienta de software librepara modelos gráficos probabilistas (2009)
2. Bottcher, S.G., Dethlefsen, C.: Learning bayesian networks with r. In: International Workshop on Distributed Statistical Computing, DSC 2003 (2003)
3. Cree, G.S., McRae, K.: Analyzing the factors underlying the structure and computation of the meaning of chipmunk, cherry, chisel, cheese, and cello (and many other such concrete nouns). Journal of Experimental Psychology: General 132, 163–201 (2003)
4. Díez-Vegas, F.J.: Teoría probabilista de la decisión en medicina (2007)
5. Fernández-Galán, S., Díez-Vegas, F.J.: Modelling Dynamic Causal Interactions with BayesianNetworks: Temporal Noisy Gates (2000)
6. Kjaerulff, U.B., Madsen, A.L.: Bayesian Networks and Influence Diagrams
7. Lacave, C.: Explicación en Redes Bayesianas (2002)

8. McRae, K., Cree, G.S., Seidenberg, M.S., McNorman, C.: Semantic feature production norms for a large set of living and non living things. Behaviour Research Methods 37, 547–559 (2005)
9. Moreno, F.J., Peraita, H.: Análisis de la estructura conceptual de categories semánticas naturales y artificiales en una muestra de pacientes de alzheimer. Psicothema 18(3), 492–500 (2006)
10. Neapolitan, R.E.: Learning Bayesian Networks. Series in Artificial Intelligence. Prentice-Hall, Englewood Cliffs (2004)
11. Nielson, T.D.: Bayesian Networks and Decision Graphs (2007)
12. Peraita, H.: Corpus lingüístico de definiciones de categorías semánticas de personas mayores sanas y con la enfermedad del alzheimer. Technical report, Departamento De Psicología Básica 1. Facultad de Psicología. UNED (2009)
13. Peraita, H., Galeote, M.Á., González-Labra, M.J.: Deterioro dela memoria semántica en pacientes de alzheimer. Psicothema 11(4), 917–937 (1999)
14. Peraita, H., Grasso, L., Mardomingo, M.C.: Análisis preliminar de rasgos de definiciones de categorías semánticas del corpus lingüístico de sujetos sanos y con enfermedad de alzheimer, Technical report, Departamento de Psicología Básica 1. Facultad de Psicología. UNED (2009)
15. Valls-Pedret, C.: Diagnóstico precoz de la enfermedad de alzheimer: fase prodrómica y preclínica. Rev. Neurol. 51(8), 471–480 (2010)

Localization and Segmentation of the Optic Nerve Head in Eye Fundus Images Using Pyramid Representation and Genetic Algorithms

José M. Molina and Enrique J. Carmona

Dpto. de Inteligencia Artificial, ETSI Informática, Universidad Nacional de Educación a Distancia (UNED), Juan del Rosal 16, 28040, Madrid, Spain
jmolina_79@hotmail.com,
ecarmona@dia.uned.es

Abstract. This paper proposes an automatic method to locate and segment the optic nerve head (papilla) from eye fundus color photographic images. The method is inspired in the approach presented in [1]. Here, we use a Gaussian pyramid representation of the input image to obtain a subwindow centered at a point of the papillary area. Then, we apply a Laplacian pyramid to this image subwindow and we obtain a set of interest points (IPs) in two pyramid levels. Finally, a two-phase genetic algorithm is used in each pyramid level to find an ellipse containing the maximum number of IPs in an offset of its perimeter and, in this way, to achieve a progressive solution to the ONH contour. The method is tested in an eye fundus image database and, in relation to the method described in [1], the proposed method provides a slightly lower performance but it simplifies the methodology used to obtain the set of IPs and also reduces the computational cost of the whole process.

1 Introduction

The location and segmentation of the optic nerve head (ONH) is of critical importance in retinal image analysis. The ONH, also called optic disk or papilla, is oval-shaped and is located in the area where all the retina nerve fibres come together to form the start of the optic nerve that leaves the back of the eyeball. There is an area without any nerve fibres called excavation (the centre of the papilla) and around it another area can be found, the neuroretinal ring, whose external perimeter delimits the papillary contour.

Support systems for the diagnosis of eye diseases based on eye fundus image information involve semi-automatic and automatic locating of the papillary contour. Specifically, the first semi-automatic strategies were based on geometric properties of the image pixels and their intensity level values [2].

Active contour-based strategies have also shown to be useful to address this problem. A characteristic property of this type of technique is that it is highly dependent on a preliminary stage of contour initialization, from which the final solution is refined. In some studies, this initialization stage has been done manually, [3,4], and in other instances automatically [5,6].

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 431–440, 2011.

Genetic algorithms (GA) have also been used efficiently as a search method to identify the papillary contour [1,7]. Other recent approaches have used different types of techniques like, for example, mathematical morphology [8], image filters [9] or the Hough Transform [10].

Basically, our method uses a Laplacian pyramid of the eye fundus image to obtain a set of interest points (IPs) in each pyramid level. Then a two-phase genetic algorithm is used to achieve a progressive solution to the ONH contour. The method here presented is a variation of the method described in [1]. The main difference between both methods lies in how to get the set of IPs. On the one hand, in the proposed method, the IPs are obtained using a Laplacian pyramid. On the other hand, in [1], they are obtained using a domain-knowledge-based method.

The article is organized as follows: Section 2 describes the methodology of the proposed method. In section 3, the obtained results are evaluated and compared with other methods. Finally, section 4 presents the conclusions and future work.

2 Method Description

The aim of this work is to locate and segment the ONH in eye fundus color photographs through the process shows in the block diagram of the figure 1. First, in order to reduce the processing computational cost, the process begins by automatically extracting a subwindow from the original image. This subwindow is approximately centered at a point of the papillary area. Then a Laplacian pyramid is applied to this image subwindow and the result is an image multiscale representation where, in each pyramid level, a set of IPs is obtained. These IPs correspond to image points of high-medium frequency, i.e., border points. We use a two-phase GA: GA2+GA1. In both phases, the goal is to search for an ellipse containing the maximum number of IPs in an offset of its perimeter. GA-2 is applied to IPs from level-2 and GA-1 is applied to ones belonging to level 1. GA-2 is run first and, when it completes its execution, the best solutions obtained are used as part of the GA-1 initial population. Finally, after running GA-1, we select the best papillary contour from the final population as the solution of our problem.

2.1 Extracting Subwindow by Gaussian Pyramid

The figure 2 shows the operator block diagram used for extracting an image subwindow from the eye fundus original image, in order to reduce the noise (distracting structures) and the processing computational cost. This operator starts applying a Gaussian pyramid of N-levels to the original image (600x400 pixels). The idea is to smooth the image intensity levels in order to eliminate the image high frequencies. In each pyramid level, the intensity of pixels belonging to the retina, blood vessels and papilla are smoothed in different intensity intervals. The smoothing process is more pronounced as the pyramid level increases. At level-N, the region of the brightest pixels (RBP) will correspond to pixels belonging to the papilla. Here we use the property that the papilla pixels have an

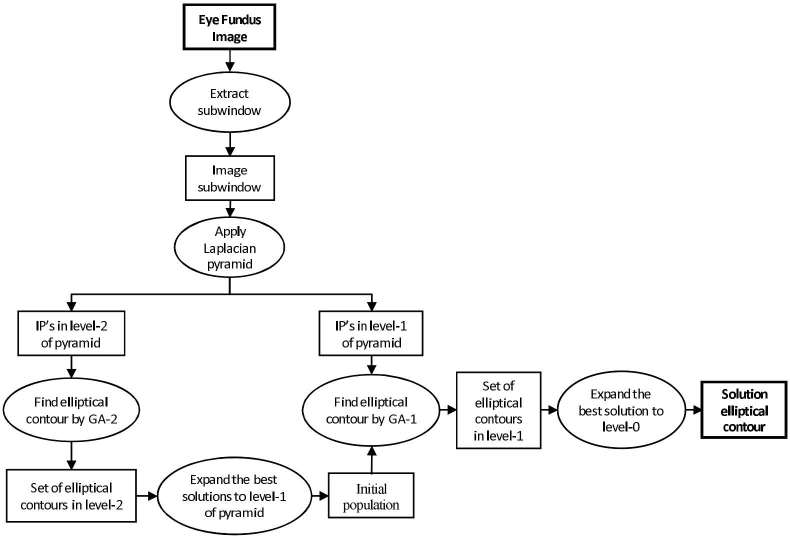

Eye Fundus Image
Extract subwindow
Image subwindow
Apply Laplacian pyramid
IP's in level-2 of pyramid
IP's in level-1 of pyramid
Find elliptical contour by GA-2
Find elliptical contour by GA-1
Set of elliptical contours in level-1
Expand the best solution to level-0
Solution elliptical contour
Set of elliptical contours in level-2
Expand the best solutions to level-1 of pyramid
Initial population

Fig. 1. Method block diagram summarizing the whole process

intensity level higher than retina and blood vessels. Next we invert the process and calculate the inverse pyramid (containing the RBP) up to the level-0 (original image size). At level-0, the centroid of the RBP is calculated and, finally, this centroid is used as central point to built the image subwindow (261x261 pixels). A similar idea is used in [11], where the pyramid is created using a simple Haar-based discrete wavelet transform. Equally, in [12], a first approximation to the pixels detection belonging to the papillary area is done by applying a threshold in each of the three RGB channels. Experimentally, the search for points belonging to the papillary area by this operator on the DRIONS database [13] has produced a 100% success. The Gaussian pyramid only was applied to the image red channel and the number of levels used was N=3. The figure 3 shows the Gaussian pyramid of an input eye fundus image, the inverse Gaussian pyramid (containing RBP) and the final subwindow.

2.2 Obtaining Interest Points by Laplacian Pyramid

The Laplacian pyramid was initially introduced by Burt&Adelson [14] in the context of compression of images. However, this technology has proved be useful in different image analysis tasks: generation and reconstruction, progressive transmission, multi-scale feature detection and enhancement. Just as we may view the Gaussian pyramid as a set of lowpass filtered copies of the original image, we may view the Laplacian pyramid as a set of bandpass filtered copies of the image. The scale of the Laplacian operator doubles from level to level of the pyramid, while the center frequency of the passband is reduced by an octave.

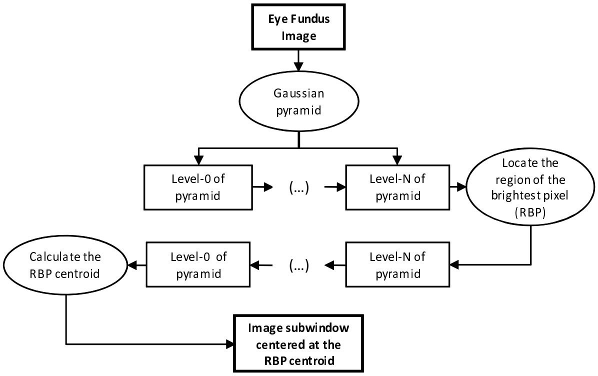

Eye Fundus Image
Gaussian pyramid
Level-0 of pyramid
(...)
Level-N of pyramid
Locate the region of the brightest pixel (RBP)
Level-N of pyramid
(...)
Level-0 of pyramid
Calculate the RBP centroid
Image subwindow centered at the RBP centroid

Fig. 2. Diagram summarizing the image subwindow extraction subtask

Thus, the Laplacian pyramid allows a representation of the original image into a hierarchy of images in such a way that each level corresponds to a different band of frequencies.

The characterization of an IP is based mainly on a property by which a point of the image which belongs to the papillary contour presents a change in its intensity level in relation to its neighboring pixels. Nevertheless, this property represents a necessary but not sufficient condition, since there may be other causes that give rise to the appearance of noise associated to other type of intensity level changes. This noise may be due, for instance, to the non-uniform color of the retina or to the existence of blood vessels. In order to reduce the image channel number, we make a weighted sum of the three RGB channels to produce a single image. Then we select as IPs the brightest pixels in the two first levels of the Laplacian pyramid applied to this average image. The figure 4 shows an example of the IPs obtained in each of one of these pyramid levels.

2.3 Obtaining Papillary Elliptic Contours by Genetic Algorithms

Genetic algorithms are one of the paradigms most frequently used in evolutionary computation and they owe their inspiration to the biological process of evolution: natural selection and survival of the fittest individuals [16]. Thus, given a problem of a specific domain, these algorithms code potential solutions using a structure of data like a chromosome (individual) where the genes are parameters of the problem proposed. The approach to the solution of each chromosome is calculated using a fitness function. Finally, by applying selection and variation operators (mutation and recombination), a population of chromosomes evolves to the optimum solution until a finalization criterion is achieved.

Since the human papillary contour is oval-shaped, this work proposes searching for the solution by approaching this contour with an ellipse, like is done in other works, e.g. [1,6,15]. From this instant, it is assumed that: (1) the geometric shape of the papilla can be approached by a non-deformable ellipse, and (2) the

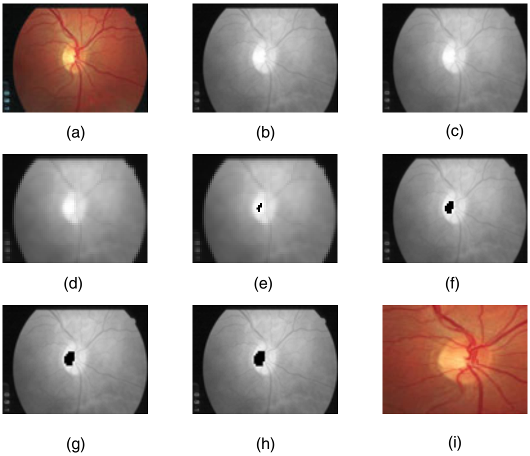


Fig. 3. An example that shows the different phases of the image subwindow extraction task: (a) original image at level 0; (b-d) image red channel Gaussian pyramid, showing levels 1, 2 and 3, respectively; (e) region of the brightest points (RBP) in level 3, represented by the centered black pixels; (f-h) inverse Gaussian pyramid (containing RBP): levels 2, 1 and 0, respectively; (i) final image subwindow centered at level-0 RMP centroid

more IPs that the ellipse contains, the more likely this ellipse will approach the real contour of the papilla. The results obtained in [1] support the validity of the two hypotheses.

Specifically, we use a two-phase GA: GA-2+GA-1 (see figure 1). GA-2 is applied to level-2 of pyramid to find an ellipse containing the maximum number of IPs in an offset of its perimeter. The result is a first approximation to our problem, i.e., a set of papillary contours in that level. Now, we expand the best contours obtained previously from level-2 to level-1 for using them as part of the GA-1 initial population. The other subset of individuals, needed to complete the GA-1 initial population, is obtained randomly. Then GA-1 is applied to IPs of level-1 with the same goal than GA-2. Finally, in order to produce the solution, the best papillary contour obtained by GA-1 is expanded from level-1 to image subwindow original level (level-0). At this point, it is important to say that we do not work in the Laplacian pyramid level-0 because the IPs obtained in this level contain a lot of noise.

The phenotypic space, solution space of the original problem, consists of the elliptic crown space defined from the infinite ellipses that can be traced in the image. To code this type of solutions, the phenotypic space is transformed into a genotypic space consisting of real vectors of five variables $[x, y, a, b, \omega]$. Thus, (x, y) represents the centre of this ellipse, (a, b) the magnitudes of its major and minor semi-axis respectively and ω the angle that its major axis forms with the x-axis. The fitness function is defined by counting the IPs number confined

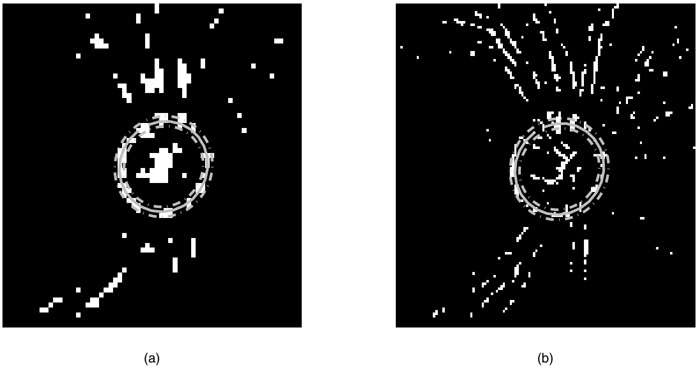


Fig. 4. Examples of sets of IPs and papillary genetic ellipses obtained: (a) Laplacian pyramid level-2; (b) Laplacian pyramid level-1. Genetic ellipses (solid lines), elliptical crowns (dashed lines)

within a concentric elliptical crown that contains the genetic ellipse. Figure 4 shows two solution ellipses adapted to the set of IPs obtained in the two levels of Laplacian pyramid: level 2 (Fig. 4a) and level 1 (Fig. 4b).

The table 1 summarizes the domain of definition of each gene used in the executions of GA-1 and GA-2. These values are obtained from [1], but they are transformed taking in count the image resolution in the pyramid levels 1 and 2. Finally, the table 2 summarizes the configuration values of each of the two AGs used in our approach.

Table 1. Definition domain (expressed in pixels) for each of the genes (variables) of a chromosome (genetic ellipse)

	GA-2	GA-1
Image Size	65x65	129x129
$[x_{min}, x_{max}]$	[15, 50]	[30, 100]
$[y_{min}, y_{max}]$	[15, 50]	[30, 100]
$[a_{min}, a_{max}]$	[8, 15]	[16, 30]
$[b_{min}, b_{max}]$	[8, 15]	[16, 30]
$[\omega_{min}, \omega_{max}]$	[15°, 175°]	[15°, 175°]

3 Experiments and Results

To measure the performance of our algorithm, we used DRIONS database [13]. In order to do the result of the evaluation quantitatively reproducible, we measured the mean discrepancy between the points of the contour obtained with the segmentation method and a gold standard defined from a contour that was the result of averaging two contours, each of them traced by an expert. Here we use

Table 2. Summary of the configuration of the two AGs used in our approach

AG Parameter	GA-2	GA-1
Initialisation	Random	Random and best solutions of GA-1
Representation	Real vector (dim: 1 X 5)	
Recombination	One-point crossover	
Mutation	Non-uniform with Gaussian distribution	
Parent selection	Stochastic uniform	
Survival selection	Generational without elite	Generational with elite

the concept of discrepancy, δ, defined in [1,6]. For visualization purposes, the ogive of discrepancy is plotted, namely, the percentage of images with discrepancy less than δ (y-axis) versus δ (x-axis).

3.1 Experiment Results

Different experiments were done in order to choose the best values related to the intrinsic parameters of our two-phase GA: population size (P), number of generations (G), elite population size (E), crossover ratio (C) and shrink value (S). The latter controls the ratio with which the mean mutation magnitude decreases and it is related to the non-uniform mutation operator with Gaussian distribution applied with a probability of one per gene. In the light of the results obtained in the experiments, the following final parameter configuration was chosen: P2=200, G2=20, C2=0.2, S2=0.8, E2=0 (for GA-2,) and P1=100, G1=50, C1=0.2, S1=0.8, E1=1 (for GA-1). The GA-1 initial population was created using 50% best solutions of the GA-2. The rest of the GA1 initial population was randomly obtained. In order to achieve a competitive result, we searched for a compromise between quality of results and computational cost.

3.2 Evaluation and Comparison of Results

In the bibliography consulted by the authors related to ONH segmentation from eye fundus color images and discarding the evaluations done subjectively and not quantitatively, we believe that the Lowell's [6] and Carmona's [1] methods are the ones that have obtained better results. Therefore, we will use these approaches as reference. Thus figure 5 shows the discrepancy curves obtained from applying the three methods to DRIONS database [13]. Specifically, the Lowell's and Carmona's curves were obtained directly from [1]. On the other hand, owing to the stochastic nature associated with the GA, the curve of discrepancy obtained by our method corresponds to the result of averaging five of them obtained as a result of executing our two-phase GA five times.

Direct comparison of the discrepancy curve obtained by Lowell's method and our method reveals two very different areas with opposite behavior, whose boundary is marked by the value $\delta = 1.4$. Below this value, Lowell's results are slightly better than our proposal, and above this value, precisely the opposite

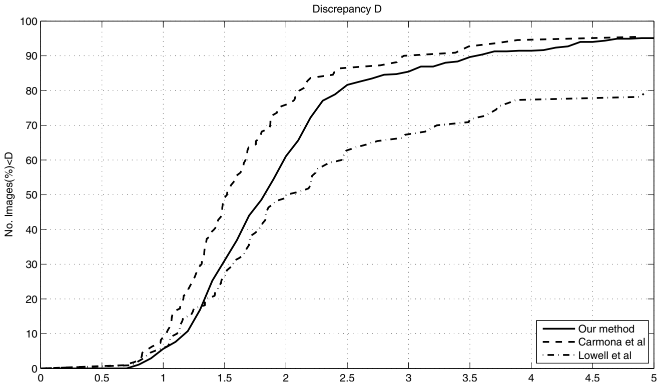


Fig. 5. Accumulated discrepancy results for our segmentaation method versus Carmona et al [1]and Lowell et al [6]

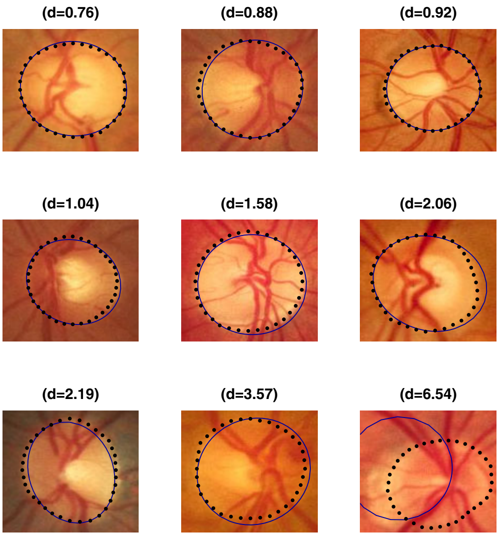


Fig. 6. Several examples of segmentation obtained with our method (solid line). The gold standard is represented by dotted line.

occurs. We think that the explanation for the results obtained in the first discrepancy subinterval $(0, 1.4)$ is that while our method approaches the solution using a non deformable ellipse, the local deformation phase in Lowell's method makes it possible to do a slight deformation of the ellipse, obtaining a better approaches the experts' real trace. However, the change in trend on the second subinterval $(1.4, 5)$ reveals the robustness of our method. In fact, while Lowell's method only obtains 80% of images below a discrepancy $\delta = 5$, our method obtains 95%.

The comparison of the discrepancy curve obtained by Carmona's method reveals that our method has a behavior slightly worse in all the discrepancy range. The explanation of this behavior could be due to the fact that the final genetic contour of our method is obtained in the Laplacian pyramid level-1, i.e, we need to expand that solution to the original image level. This fact could also explain why the gap between both discrepancy curves is more pronounced for low levels of discrepancy and less pronounced for high values. In fact, for a discrepancy value of $\delta = 5$, both methods match their performance. On the other hand, the computational cost of our method is lower: 9.000 evaluations of the fitness function against 20.000 evaluations used by Carmona's method.

Finally, the figure 6 shows different examples of papillary contours obtained in this work and they are compared with the experts' gold standard.

4 Conclusions and Future Work

We have described a new automatic method to locate and segment the ONH in eye fundus color photographic images. This method is inspired in the approach presented in [1]. The performance of our method was compared with two competitive methods in the literature.

In relation to the results obtained in [1], we obtain a slightly lower performance. In contrast, our method has shown to be faster. Also, the way of obtaining our set of IPs is simpler: our fitness function only pays attention to the number of IPs inside the elliptic crown, i.e., it is independent of the domain knowledge. Finally, we can also say that our method outperforms the results obtained by [6] in most of the discrepancy domain ($\delta > 1.4$).

A disadvantage of our method is that we cannot work at Laplacian pyramid level-0 because the set of IPs obtained in that level contains a lot of noise. Therefore, we have to expand directly the genetic solution obtained from level-1 to level-0, but this operation seems to produce a lack of accuracy. We propose as future work to investigate other types of multi-resolution image representation techniques like wavelet transform or apply images filter directly in the input image (level-0) in order to obtain sets of IPs with a low level of noise.

Acknowledgment. This work was supported in part by funds of the Advanced Artificial Intelligence Master Program of the Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain.

References

1. Carmona, E.J., Rincón, M., García-Feijoo, J., Martínez-de-la-Casa, J.M.: Identification of the Optic Nerve Head with Genetic Algorithms. Artificial Intelligence in Medicine 43(3), 243–259 (2008)
2. Cox, J., Wood, I.: Computer-assisted optic nerve head assessment. Ophthalmic and Physiological Optic 11(1), 27–35 (1991)

3. Morris, T., Donnison, C.: Identifying the neuroretinal rim boundary using dynamic contours. Image and Vision Computing 17(3-4), 169–174 (1999)
4. Mendels, F., Heneghan, C., Thiran, J.: Identification of the optic disk boundary in retinal images using active contours. In: Proceedings of the Irish Machine Vision and Image Processing Conference (IMVIP 1999), pp. 103–115 (1999)
5. Osareh, A., Mirmehdi, M., Thomas, B., Markham, R.: Colour morphology and snakes for optic disc localisation. In: Proceedings of the 6th Medical Image Understanding and Analysis Conference, pp. 21–24 (2002)
6. Lowell, J., Hunter, A., Steel, D., Basu, A., Ryder, R., Fletcher, E., Kennedy, L.: Optic nerve head segmentation. IEEE Transaction on Medical Imaging 23(2), 256–264 (2004)
7. Novo, J., Penedo, M.G., Santos, J.: Localisation of the optic disc by means of GA-optimised Topological Active Nets. Image and Vision Computing 27(10), 1572–1584 (2009)
8. Welfer, D., Scharcanski, J., Kitamura, C.M., Dal Pizzol, M.M., Ludwig, L.W.B., Ruschel, D.: Segmentation of the optic disk in color eye fundus images using an adaptive morphological approach. Computers in Biology and Medicine 40, 124–137 (2010)
9. Rangayyan, R.M., Zhu, X., Ayres, F.J., Ells, A.L.: Detection of the Optic Nerve Head in Fundus Images of the Retina with Gabor Filters and Phase Portrait Analysis. Journal of Digital Imaging 23(4), 438–453 (2010)
10. Zhu, X., Rangayyan, R.M., Ells, A.L.: Detection of the Optic Nerve Head in Fundus Images of the Retina Using the Hough Transform for Circles. Journal of Digital Imaging 23(3), 332–341 (2010)
11. Lalonde, M., Beaulieu, M., Gagnon, L.: Fast and Robust Optic Disc Detection Using Pyramidal Decomposition and Hausdorff-Based Template Matching. IEEE Transactions on Medical Imaging 20(11), 1193–1200 (2001)
12. Bachiller, M., Rincón, M., Mira, J., García-Feijó, J.: An automatic system for the location of the optic nerve head from 2D images. In: Mira, J., Prieto, A.G. (eds.) IWANN 2001. LNCS, vol. 2085, pp. 319–327. Springer, Heidelberg (2001)
13. DRIONS-DB: Digital Retinal Images for Optic Nerve Segmentation Database, http://www.ia.uned.es/personal/ejcarmona/DRIONS-DB.html (accessed on August 21, 2009)
14. Burt, P.J., Adelson, E.H.: The Laplacian Pyramid as a Compact Image Code. IEEE Transactions on Communications 31(4), 532–540 (1983)
15. Shields, M.B.: Color atlas of glaucoma. Williams and Wilkins, Baltimore (1998)
16. Holland, J.H.: Adaption in Natural and Artificial Systems. MIT Press, Cambridge (1992)

A Multisensory Monitoring and Interpretation Framework Based on the Model–View–Controller Paradigm

José Carlos Castillo[1], Angel Rivas-Casado[2], Antonio Fernández-Caballero[1,3], María T. López[1,3], and Rafael Martínez-Tomás[2]

[1] Instituto de Investigación en Informática de Albacete (I3A), n&aIS Group,
Campus Universitario s/n, 02071-Albacete, Spain
{JoseCarlos.Castillo,Antonio.Fdez,Maria.LBonal}@uclm.es

[2] Departamento de Inteligencia Artificial, E.T.S.I. Informática,
Universidad Nacional de Educación a Distancia, 28040-Madrid, Spain
{arivas,rmtomas}@dia.uned.es

[3] Departamento de Sistemas Informáticos, Universidad de Castilla-La Mancha,
Campus Universitario s/n, 02071-Albacete, Spain

Abstract. This paper proposes a monitoring and interpretation framework inspired in the Model–View–Controller (MVC) paradigm. Indeed, the paper proposes the extension of the traditional MVC paradigm to make it more flexible in incorporating the functionalities of a monitoring and interpretation system. The proposed model is defined as a hybrid distributed system where remote nodes perform lower level processing as well as data acquisition, while a central node is in charge of collecting the information and of its fusion. Firstly, the framework levels as well as their functionalities are described. Then, a fundamental part of the proposed framework, namely the common model, is introduced.

1 Introduction

Monitoring and interpretation systems have to understand and predict actions of the objects present in the scenario. To date there are many approaches developed for public transport monitoring, such as airports [14], ports [12], railway and subway stations [7,13], and traffic control [1]. Other important monitored public places are banks, shops, home or parking lots [15]. There are also systems indicated for human activity monitoring [11,4] or simply as an answer to industrial needs [5].

Moreover, the requirements of surveillance systems have led to systems able to monitor large areas. They go beyond simple object detection, including tracking and activity detection, and involving several sensors. Most works are centered in the combination of a set of cameras for this purpose. In [16] a system for classification and tracking of soccer players is proposed. The system consists of eight cameras (each one with its own processor) and a central processor for data collection and fusion. Another approach [6] describes the "EasyLiving" system.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 441–450, 2011.

The system performs multicamera tracking for behavior recognition in homes. Moreover, [18] introduces a client-server architecture for detection, recognition and tracking of people and vehicles. Finally, in [9] a distributed surveillance system is described.

The monitoring and interpretation framework proposed in this paper arises from the *Model–View–Controller* (MVC) paradigm [10]. This paradigm explicitly separates user inputs, world models and visual feedback, and proposes three kinds of blocks to handle them. In first place, the *model* is in charge of managing the application data as well as performing object initialization, providing information about the application state and primitives to update the state. On the other hand, the *view* provides a world representation adapted to the user's needs. And, finally, the *controller* collects all system inputs, invoking the model primitives to update its objects.

2 Extension of MVC for Monitoring and Interpretation Tasks

The MVC paradigm has been widely used for web programming where the model accesses the data (Database, XML), the view defines the user interface (HTML + CSS), and the controller reacts to the events by modifying the view and the model (PHP, .NET, JAVA). Despite the paradigm seems to be appropriate for the mentioned applications, its usage in monitoring and interpretation tasks entails a few improvements to the original paradigm. Therefore, the current work proposes the extension of the traditional MVC paradigm to make it more flexible in incorporating the functionalities of a monitoring and interpretation system. The proposed extensions also allow that already developed algorithms can be incorporated to the framework without great design changes. For that reason, business logic is separated from the model, generating a newer execution block managed by the controller. The new block is named *algorithm* and its operation corresponds to the algorithms of each framework level. Thus, the framework holds a series of extended MVC modules, each one devoted to a different level of the framework. Moreover, the model functionality in the proposed extension is the management of application data through a series of primitives (see Fig. 1).

The previously described modules are integrated into the traditional architecture through the incorporation of a new module. This module is the base where the remaining levels of the architecture are placed. In this special MVC module, the new model block composes the *common model.* This common model can be considered as the backbone of the architecture, since it holds the parameters needed or generated by the different levels of the architecture. Thereby, the dependencies among modules are removed and the data access is simplified by providing a unique module with well defined inputs and outputs. Later, in section 4, the common model features are defined, as well as the parameters at each architecture level. The view has been modified too in order to contain every interface of the different modules that compose the architecture levels. Thereby,

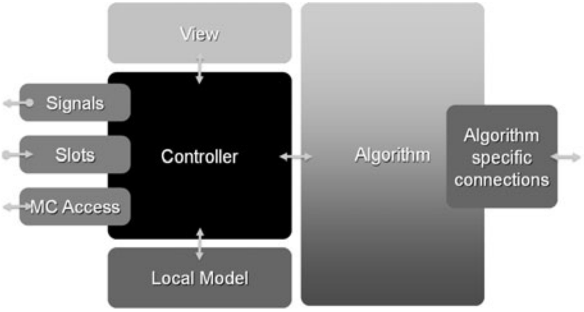


Fig. 1. Extension to the traditional MVC

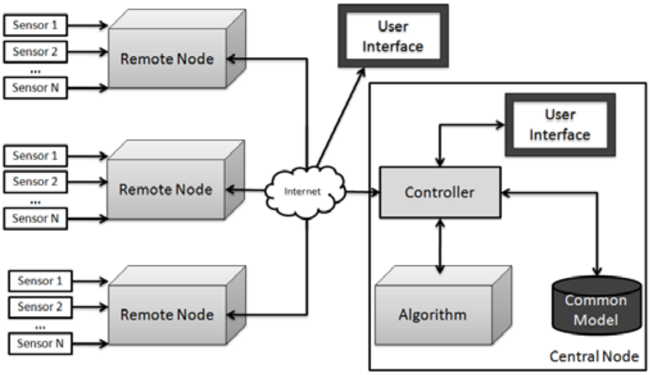


Fig. 2. Execution model. Hybrid system.

this module's view is defined as a multiple document interface (MDI) form where each view of the remaining modules (levels) appear as forms within the main view. In this module, the controller's function is to manage the execution of the remaining modules, monitoring their correct operation.

3 Definition of the Framework

Prior to detailing each system level, it is necessary to describe the execution model. It is defined as a hybrid distributed system where remote nodes perform lower level processing as well as data acquisition, while a central node is in charge of collecting the information and of its fusion. In Fig. 2 a schematic representation of the framework modules is shown. Remote modules have the MVC extended structure described in the previous section, but perform just a subset of all architecture layers described in the next section. Common model, a global control and view are also held in the central node, with MVC extended structure too.

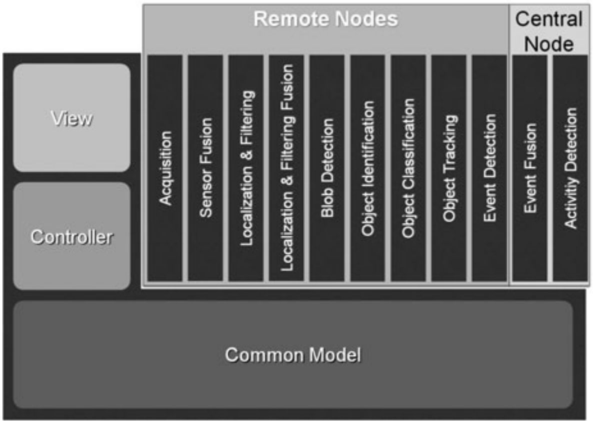


Fig. 3. Framework layers

3.1 Layers of the Framework

The current section describes the framework levels (see Fig. 3). In first place, the common model stands out. As aforesaid, this block is accessible from every level. The levels are composed by MVC extended modules (see Fig. 1). Next, the functionality of each level is described, although, in order to keep the framework as generic as possible, no algorithm is associated. Of course, the proposed levels are just a guideline to create the framework, but it is possible to include new levels according to the application requirements.

Acquisition: This level directly interacts with the digital analog devices, measuring from the physical world and adapting these measures to be usable by the system. The measures are data from the sensors as well as data from other information sources (disk, database, and so on). The acquisition level also performs information preprocessing.

Sensor Fusion: This level is in charge of merging the sensor data to improve the information quality (more complete and accurate). Fusion algorithms may also operate with different spectrum images and are capable of introducing knowledge on the domain.

Localization and Filtering: The third framework level is dedicated to isolate the objects of interest contained in the input images. This level may hold a wide range of methods, from the simplest one (a binarization applied to infrared (IR) images [2]) to other more complex approaches yielding better results [8]. On the other hand, this level also filters the spots corresponding to the objects of interest with the aim of eliminating possible noise.

Localization and Filtering Fusion: This level fuses images obtained in localization and filtering stage as there might be several localization and filtering approaches running in the framework (e.g. one devoted to color images and another to IR images). Thus, this level seeks for the most benefic features from the input images.

Blob Detection: The blob detection level filters isolated spots misdetected in the previous levels. Besides, the blob detection level is in charge of extracting information associated to the spots to allow a more efficient analysis of the objects. This information is application-dependent.

Object Identification: This level operates with objects instead of blobs. This enhances the information abstraction, mapping object coordinates into the real world instead of simply operating with image coordinates.

Object Classification: This level is specially important to perform a good activity analysis because it provides knowledge about "what" the object is. Also, object classification may provide information about the objects' orientation.

Object Tracking: This level is in charge of mapping the image objects' coordinates into the real map. Thus, it calculates the trajectories followed by the moving objects within the scenario, independently of the particular sensor that detected them. It also makes predictions about future positions of the objects on the basis of the previously detected trajectories. This level uses the information from the common model referring to the map, the sensors situation and its coverage range.

Event Detection: The event detection level generates semantic information related to the behavior of the objects in the scenario. These events are considered instantaneous, that is, they are time independent. Some examples are events such as running, walking or falling, which can be detected with just one or at most a few input images. This is the last layer of the framework held within remote nodes (see Fig. 2). The next layers are implemented in the central node together with the common model and the central controller and view.

Event Fusion: In a multisensory monitoring and interpretation system, where several sensors monitor a common scenario, the events generated from different sources usually do not match. This is why the event fusion level is necessary to unify the information arriving from the different sensory data generated in the previous level.

Activity Detection: This final level of the architecture is in charge of the analysis and detection of activities already associated to temporal features. After event fusion, the current level has a better knowledge of what is happening in the scenario according to the detected events. Hence, the activities detected at this level can be translated into actions along the scenario, providing a higher abstraction level.

4 Definition of the Common Model

After describing the framework levels, it is time to simplify the information exchange among them. For this reason, a new and fundamental layer is considered. This layer, known as the common model, gathers all the information from the different levels while providing primitives to access the information. The common model introduced is a variation of the traditional model of the MVC, where the algorithm of each module processes the information – always under the controller's management. Thereby, the common model is only in charge of holding the common information to be accesses by every execution module. For this purpose, the primitives that allow managing the data are provided. To properly define the common model, we will start with the layers that compose the architecture. Since the input and output parameters are known, it is possible to estimate which of them belong to the common model.

Acquisition: This level obtains data from diverse sources (camera, sensors, database, etc.) which determine the nature of the parameters contributing to the common model. In first place, the common model counts on a list of images captured from the cameras, regardless of the subjacent camera technologies (IR, color, range, etc.), adapted to the application requirements. A time parameter (ID_TIME) is associated to each image to ensure a correct synchronization. Moreover, sensor readings are also included in the common model as XML structures (again with an associated time parameter).

Sensor Fusion: As the most common parameters in a monitoring and interpretation system are images, this level contributes to the common model with a list of new fused images, independent from the acquisition image list. The content of this list depends on the fusion algorithms implemented . Again, time information is associated to fused images. Fusion of non-visual sensory data is open due to their high dependence from the sensor technology and the application.

Localization and Filtering: In literature, it is common to mix two terms when talking about segmentation [3,17]: Firstly, there is localization and filtering defined as the process whereby, from an input image, a set of spots containing the objects of interest are isolated. And, secondly, there is the blobs detection process. Clearly, this level receives images as input coming either from acquisition level or from sensor fusion level. Output parameters are a list of images containing isolated regions. These regions are highlighted from the background (white foreground on black background). Again, it is important to provide time information to the images. Thus, a time field (ID_TIME) is again attached to the images.

Localization and Filtering Fusion: This level fuses the information coming from the previous one. This is because several algorithms can be incorporated to the framework at localization and filtering level and its combination should provide a more accurate localization. This level incorporates a new fused image list, containing the results of its operation, to the common model.

Blob Detection: This level operates with blobs instead of sensor measures. The detection process uses information from the localized and filtered images to obtain the blobs contained. As some kinds of sensors provide distance information (e.g. range sensors), blob coordinates are defined with six components (x_{image},y_{image},z_{image},$width_{image}$,$height_{image}$,$depth_{image}$), even though depth components (z_{image},$depth_{image}$) might be void. During the blob detection process, other parameters defining the spots, such as the contour, brightness or color information may also be extracted. On the other hand, heuristic methods for detection can be applied to discard wrong spots. Again, time information and a unique identifier are associated to each blob.

Object Identification: At this level, object-level information is obtained from the blobs information. This level takes into account the history of the objects in the scene, that is, object features are updated along time by comparing blob information from current and previous iterations. Thanks to the knowledge about the history of an object, it is possible to calculate parameters derived from its motion (e.g. direction and speed). Also, information regarding corners or invariant points can be added to the object's definition, depending on the application needs.

Object Classification: The classification level uses contour information for clipping the images (acquisition or fusion level) as inputs to the classifier. This way, the classification algorithm provides information about what the object (its class) is. Classification methods can also obtain the object's orientation, which is useful for the activities detection. This level utilizes as input the acquired images as well as the information generated from the previous level (contours, invariant points, and so on).

Object Tracking: This level is in charge of calculating the trajectories followed by the moving objects in the scene. Previously it is necessary to calculate the objects' positions in the real world's map. On the other hand, a monitoring and interpretation system must keep tracking the objects, independently of which sensor is detecting it. For that reason, the calculated trajectories must be independent from the sensors (if possible).

Event Detection: This level translates the information coming from the lower levels into semantic primitives to model the relationships involving the objects present in the scenario. Thus, the inputs are the objects tracked by the previous levels and the sensor data whilst the output is closely linked to the event detection algorithm (HMM, SVM, Bayesian networks, etc.). Anyway, it is possible to define a common event representation format by using the flexibility of XML abstractions (see Table 1). XML provides an open structure, able to wrap all proposal outputs, homogenizing them for use by upper layers. This representation also allows managing the probability associated to the events. Again, a time parameter is associated to the events to simplify the event fusion process in the next level.

Table 1. Event structure

```
<events>
    <event name=Run>
        <object>1</object>
        <speed>5</speed>
        <probability>0.7</probability>
        <timeStamp>1256953732</timeStamp>
    </event>
    <event name=Walk>
        <object>2</object>
        <speed>1</speed>
        <probability>0.6</probability>
        <timeStamp>1256953735</timeStamp>
    </event>
</events>
```

Table 2. Main features of the common model

Level	Parameters	Details
Acquisition	- Acquired image list - Data from sensors ⇒ XML - IdTime	Data collection Information sources management Synchronization
Sensor Fusion	- Fused image list	Create new images from the acquired ones
Localization & Filtering	- Localization & Filtering image list	Isolate spots containing the objets of interest
Localization & Filtering Fusion	- Fused image list	Merge localized & filtered images
Blob Detection	Blob list containing: - IdTime - Contours - Box (X,Y,Z,W,H,D) - Color and brightness	From pixel-level information to blob-level Obtain parameters to characterize the spots (coordinates, color info, contours, etc.)
Object Identification	Object list containing: - IdTime - Box (X,Y,Z,W,H,D) - Real position - Label - Contours - Characteristic Points - Direction - Speed	Blob-level information to object level Information remain among iteration, being updated Calculation of parameters defining object motion Mapping of the image coordinates of the objects into the real world
Object Classification	- Class - Orientation	Update object list Provide information concerning "what" the objects are and their orientation
Object Tracking	- Trajectory	Calculation of object trajectories Match objects changing the sensors' detection field
Event Detection	- Event list	Instantaneous event detection (semantic primitives) Events are wrapped into XML structures
Event Fusion	- Fused event list	Performed by central node Update event list, discarding mismatching events and unifying repeated ones Synchronization is a key aspect (when to fuse) Feedback to local nodes according to their received events
Activity Detection	- Activity list	Global view of the scenario Spatial and temporal information
Scenario Modeling	Scenario map - Light conditions - Temperature - Sensors and their ranges - Detection distance	Related sensors information Global and local modeling

Event Fusion: This event fusion level allows the consideration of new information sources (the remote nodes), even though, at this level, sources provide events. Event fusion provides a general view of the scenario by merging all events, and eliminating wrong events. This again brings up the necessity for a good synchronization of the nodes. The operation of this level generates an event structure (as shown in Table 1) used as input by the next level.

Activity Detection: The higher level of the framework is devoted to the detection of activities from the events detected in the previous levels. As the events are generated by remote (distributed) nodes, their fusion provides a general view of the activities carried out in the scenario. This level provides the common model with high level activities detected that may involve several sensors in the scenario and several objects.

Scenario Modeling: Even though scenario modeling does not appear as a level within the framework definition, it is a key aspect that enables the relations among sensor information for situating the objects in the scenario. Scenario modeling can be seen as a two stage modeling. On the one hand, there is a global modeling that includes aspects such as map definition, map calibration, environmental condition setting (ambient light or temperature), and sensor placement and its range. On the other hand, there exists a modeling that is sensor-specific like the camera field of view. Some parameters are dynamically updated according to the sensed values. Table 2 summarizes the main features of the architecture levels, as well as the parameters provided to the common model.

5 Conclusions

This paper has introduced a monitoring and interpretation framework inspired in the MVC paradigm. The paper has proposed the extension of the traditional MVC paradigm for the inclusion of the functionalities of a monitoring and interpretation system. The proposed model is defined as a hybrid distributed system where remote nodes perform lower level processing as well as data acquisition. A a central node is in charge of collecting the information and of its fusion. The remote nodes hold the following layers of the framework: acquisition, sensor fusion, localization and filtering, localization and filtering fusion, blob detection, object identification, object classification, object tracking, event detection, event fusion. The central nodes incorporates activity detection and scenario modeling.

Acknowledgements

This work was partially supported by Spanish Ministerio de Ciencia e Innovación TIN2010-20845-C03-01 and TIN2010-20845-C03-02 grants, and by Junta de Comunidades de Castilla-La Mancha PII2I09-0069-0994 and PEII09-0054-9581 grants.

References

1. Celik, T., Kusetogullari, H.: Solar-powered automated road surveillance system for speed violation detection. IEEE Transactions on Industrial Electronics 57(9), 3216–3227 (2010)
2. Fernández-Caballero, A., Castillo, J.C., Serrano-Cuerda, J., Maldonado-Bascón, S.: Real-time human segmentation in infrared videos. Expert Systems with Applications 38(3), 2577–2584 (2011)

3. Fernández-Caballero, A., Castillo, J.C., Martínez-Cantos, J., Martínez-Tomás, R.: Optical flow or image subtraction in human detection from infrared camera on mobile robot. Robotics and Autonomous Systems 58(12), 1273–1280 (2010)
4. Gascueña, J.M., Fernández-Caballero, A.: Agent-oriented modeling and development of a person-following mobile robot. Expert Systems with Applications 38(4), 4280–4290 (2011)
5. Geradts, Z., Bijhold, J.: Forensic video investigation. In: Multimedia Video Based Surveillance Systems, ch. 1, pp. 3–12 (2000)
6. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., Shafer, S.: Multi-camera multi-person tracking for easyliving. In: Proceedings of the Third IEEE International Workshop on Visual Surveillance, pp. 3–10 (2000)
7. Lo, B.P.L., Sun, J., Velastin, S.A.: Fusing visual and audio information in a distributed intelligent surveillance system for public transport systems. Acta Automatica Sinica 29(3), 393–407 (2003)
8. Moreno-Garcia, J., Rodriguez-Benitez, L., Fernández-Caballero, A., López, M.T.: Video sequence motion tracking by fuzzification techniques. Applied Soft Computing 10(1), 318–331 (2010)
9. Nguyen, N.T., Venkatesh, S., West, G., Bui, H.H.: Multiple camera coordination in a surveillance system. ACTA Automatica Sinica 29(3), 408–422 (2003)
10. Reenskaug, T.: Thing-model-view-editor - an example from a planning system (1979), http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html
11. Rivas-Casado, A., Martínez-Tomás, R., Fernández-Caballero, A.: Multiagent system for knowledge-based event recognition and composition. Expert Systems: The Journal of Knowledge Engineering (2011) (in press)
12. Rodriguez, M.D., Shah, M.: Visual surveillance in maritime port facilities. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6978, pp. 697811–697818 (2008)
13. Ronetti, N., Dambra, C.: Railway station surveillance: the Italian case. In: Multimedia Video Based Surveillance Systems, ch. 1, pp. 13–20 (2000)
14. White, I.H., Crisp, M.J., Penty, R.V.: A photonics based intelligent airport surveillance and tracking system. In: Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 11–16 (2010)
15. Wong, W.K., Lim, H.L., Loo, C.K., Lim, W.S.: Home alone faint detection surveillance system using thermal camera. In: Second International Conference on Computer Research and Development, pp. 747–751 (2010)
16. Xu, M., Orwell, J., Lowey, L., Thirde, D.: Architecture and algorithms for tracking football players with multiple cameras. IEE Proceedings - Vision, Image and Signal Processing 152(2), 232–241 (2005)
17. Yin, Z., Collins, R.: Moving object localization in thermal imagery by forward-backward motion history images. In: Augmented Vision Perception in Infrared, pp. 271–290 (2009)
18. Yuan, X., Sun, Z., Varol, Y., Bebis, G.: A distributed visual surveillance system. In: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 199–204 (2003)

Agent-Based Development of Multisensory Monitoring Systems

José Manuel Gascueña[1], Antonio Fernández-Caballero[1,2],
Elena Navarro[1,2], Juan Serrano-Cuerda[1], and Francisco Alfonso Cano[1]

[1] Instituto de Investigación en Informática de Albacete (I3A), n&aIS Group,
Campus Universitario s/n, 02071-Albacete, Spain
{JManuel.Gascuena,Juan.SerranoCuerda,Francisco.Cano}@uclm.es
[2] Departamento de Sistemas Informáticos, Universidad de Castilla-La Mancha,
Campus Universitario s/n, 02071-Albacete, Spain
{Antonio.Fdez,Elena.Navarro}@uclm.es

Abstract. This paper introduces the use of the *VigilAgent* agent methodology to develop monitoring systems. This work is based on the suitability of the specific characteristics of agency for developing monitoring systems. It is usual to develop them following an ad-hoc approach instead of using a methodology for achieving quality standards expected from commercial software. In this paper, the five phases of *VigilAgent*, namely System specification, Architectural Design, Detailed design, Implementation and Deployment, are introduced. The proposal is validated through the case study of controlling the access of human beings to a specific area.

1 Introduction

Security systems [9] are being installed in environments such as bank, parking, motorway and underground to protect humans from attacks or burglaries. The development of monitoring systems is very complex as they work in highly dynamic and heterogeneous environments. They deploy, in the observed scenarios, several kinds of sensors that perform actions with a certain degree of autonomy to collect information about their surrounding area, and to cooperate in the recognition of special situations in a semi-automatic way. The characteristics of autonomy and cooperation are often cited as the rationale of why multi-agent systems (MAS) are especially appropriate for monitoring tasks [8], [6], [10]. In fact, agent technology has already been used in several monitoring systems [4]. However, to the best of our knowledge, they are usually developed following an ad-hoc approach without using a methodology that guides stakeholders in achieving quality standards expected from commercial software. So, this paper proposes the introduction of an Agent-Oriented Software Engineering (AOSE) methodology, named *VigilAgent*, to carry out well documented monitoring applications throughout the different phases that form the development process.

In this paper, the *VigilAgent* methodology is theoretically introduced and also applied to face a particular monitoring task supported in these systems. In

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 451–460, 2011.

this case *VigilAgent* is used to control the access to a specific area. Assigning human personal to carry out this task is a common solution. However, this is an expensive solution where human's attention decreases after some elapsed time working or watching video. Thus, it sounds interesting to perform this task automatically via software which is responsible for collecting information to inform when some anomalous situation occurs. In addition, it is relevant to record normal situations to guarantee some usual activities; for instance, that the current occupation capacity does not exceed above a threshold. So, a company takes knowledge about economical losses arising for cunning users, where it is necessary to have more personal to prevent risky situations. The cost of the personnel is decreased because it is not necessary to have a guard in every access control point. Notice that monitoring software does not replace the security personnel but is complementary to offer a more efficient solution.

The rest of the paper is organized as follows. In section 2, an overview of *VigilAgent* methodology phases is offered, justifying why Prometheus [5], INGENIAS [7] and ICARO-T [3] technologies are integrated. Then, section 3 introduces the case study used to demonstrate the applicability of *VigilAgent*. Finally, section 4 offers some conclusions.

2 Description of the VigilAgent Methodology

The five phases of *VigilAgent* are briefly described next. (1) *System specification* - the analyst identifies the system requirements and the environment of the problem, which are obtained after several meetings arranged with the client; (2) *Architectural design* - the system architect determines what kind of agents the system has and how the interaction between them is; (3) *Detailed design* - the agent designer and application designer collaborate to specify the internal structure of each entity that makes up the system overall architecture produced in the previous phase; (4) *Implementation* - the software developer generates and completes the application code; and (5) *Deployment* - the deployment manager deploys the application according to a specified deployment model.

At this point, it is worthy of notice several issues about this development process. The first one is that phases named system specification and architectural design in *VigilAgent* are the two first phases of Prometheus methodology [5]. Another detail is that the third phase of *VigilAgent* (detailed design) uses models of INGENIAS [7]. Finally, notice that code is generated and deployed for ICARO-T framework [3]. Several reasons that are introduced in the following have conducted to this integration.

Prometheus is significant because of the guidelines it offers to identify which the agents and their interactions are. Another advantage of Prometheus is the explicit use of the concept *scenario* which is closely related to the specific language used in the monitoring domain. Indeed, a monitoring application is developed to deal with a collection of scenarios. Nevertheless, notice that Prometheus last phase has not been integrated in *VigilAgent* because it focuses on belief-desire-intention (BDI) agents and how the entities obtained during the design are transformed into concepts used in a specific agent-oriented programming language named JACK [1]; this supposes, in principle, a loss of generality. On the contrary, INGENIAS does facilitate a general process to transform models specified during the design phase into executable code. However, INGENIAS does not offer guidelines to identify the entities of the model; the developer's experience is necessary for their identification. Therefore, *VigilAgent* methodology is not developed from scratch but integrates facilities of both Prometheus and INGENIAS to take advantage of both of them.

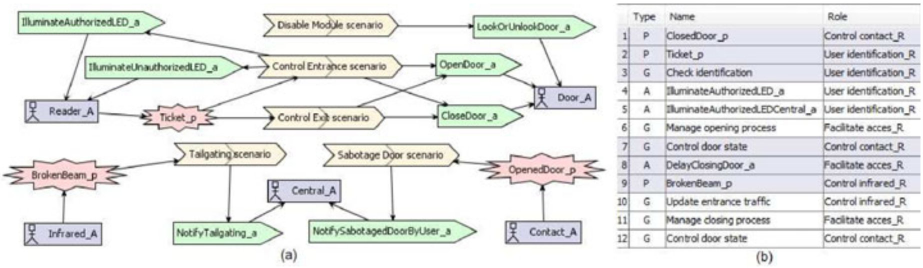
Regarding implementation, the ICARO-T framework has been selected because it provides high level software components that facilitate the development of agent applications. Moreover, it is independent of the agent architecture; that is, the developer can develop new architectures and incorporate them in the framework. This is a clear difference regarding other agent frameworks such as JACK or JADE [1], which provide a middleware, instead of an extensible architecture, to establish the communications among agents. An additional advantage are the functionalities already implemented in the framework to automatically carry out component management, application initialization and shutdown, reducing the developers' amount of work and guarantying that the components are under control. These last functionalities are usually not provided by other frameworks.

3 A Case Study: Access Control

Access control is the usual and basic term used for monitoring and controlling entrances to and exits from a specific area. This section illustrates how to use *VigilAgent* in order to develop an intelligent system that automatically controls entrances/exits of humans to/from an enclosure throughout the installed modules. Specifically, each module facilitates the entrances and exits according to its configuration and is composed of the following components: a reader device, an automatic door, a contact sensor and an infrared sensor.

In order to go in/out of the enclosure throughout a module, first, the user inserts a ticket into the reader device that the system verifies against the users' database. Then, a LED illuminates in green if the user is authorized, otherwise it illuminates in red. Moreover, if the user is authorized then the door is opened, and closed once the user has crossed or some time has elapsed.

In addition, the system collects and shows the guard statistics about the number of humans crossing each door and the number of humans located inside the enclosure by using the infrared sensor located in each module. It should also control if any anomalous situation happens, such as tailgating or if a door is blocked by a human when the system opens it. A *tailgating* situation is detected when some cunning human crosses a door that has been opened by a user correctly authenticated. The system also shows the state of the devices and offers the guard the possibility of disabling a module if its door remains closed despite having correctly authenticated a user.



	Type	Name	Role
1	P	ClosedDoor_p	Control contact_R
2	P	Ticket_p	User identification_R
3	G	Check identification	User identification_R
4	A	IlluminateAuthorizedLED_a	User identification_R
5	A	IlluminateAuthorizedLEDCentral_a	User identification_R
6	G	Manage opening process	Facilitate acces_R
7	G	Control door state	Control contact_R
8	A	DelayClosingDoor_a	Facilitate acces_R
9	P	BrokenBeam_p	Control infrared_R
10	G	Update entrance traffic	Control infrared_R
11	G	Manage closing process	Facilitate acces_R
12	G	Control door state	Control contact_R

Fig. 1. (a) Analysis overview diagram; (b) Steps of Control Entrance scenario

3.1 System Specification Phase

Usually, the system specification phase begins with the *analysis overview diagram*, which shows the interactions between the system and the environment (see Fig. 1a). At this level, firstly, an *actor* for each device of a module (reader, door, infrared and contact sensors) has been identified; there is also a *Central_A* actor representing the user interface that supports the human interaction with the system, that is, it shows the monitored activities to the security guard, and the commands he/she can send to the system to disable modules. Moreover, on the one hand, the information that comes from the environment is identified as *percepts*. For example, the code associated to the card introduced by the user into the reader (*Ticket_p*) and the signal captured by the infrared device when its beam is broken (*BrokenBeam_p*). On the other hand, every operation performed by the system on the actors is identified as an *action*. For example, the commands issued to open and close a door (*OpenDoor_a* and *CloseDoor_a*) and alert messages displayed on the user interface to notify that an anomalous situation happens (*NotifySabotagedDoorByUser_a* and *NotifyTailgating_a*). Finally, relations with the scenarios identified to control the entrance, the exit and anomalous situations are established (see *Tailgating* and *Sabotage Door* scenarios).

A scenario is a sequence of structured steps - labeled as action (A), percept (P), goal (G), or other scenario (S) - that represents a possible execution way of the system. As an example, Fig. 1b illustrates the process performed by the system to control the crossing of authorized users through a module. This scenario begins when it is perceived that the door is closed (step 1) and the user introduces his/her identification ticket into the reader (step 2). Then, the user database is queried to check if this ticket belongs to an authorized user (step 3). Based on this information, the LEDs are illuminated to notify which result is obtained with this checking (steps 4 and 5). After, the door is opened (step 6), the door state is supervised (step 7), a delay is introduced allowing the user to cross the module before the door is closed again (step 8), it is perceived when the infrared sensor is interrupted (step 9), the counter of people that cross the module is updated (step 10) and the process to close is started (step 11). The scenario finishes monitoring the door state (step 12).

Table 1. Roles description

Role	Description
User identification_R	It manages the user identification process to gain access to or come out of enclosure.
Facilitate access_R	It manages the door actuator to provide the users authenticated correctly with access.
Control infrared_R	It aims to update statistics when the infrared sensor detects a person located in front of it.
Control contact_R	Its objective is to monitor the door state thanks to the information perceived by the contact sensor installed in the door.
Management tailgating_R	It is responsible for detecting and notifying when the anomalous situation named tailgating happens.
Management sabotage door_R	It is responsible for detecting and notifying when someone, different from the security guard, blocks the door.
Intercommunication_R	It is responsible for managing the communication between software entities for person access control.

For every scenario a goal is identified that represents the goal to be achieved by the scenario. In the proposed multi-agent system approach, several agents communicate and coordinate to pursue the common general goal *Control entrance and exit*. In the *goal overview diagram*, this general goal has been refined into five goals (*Control Entrance*, *Control Exit*, *Tailgating*, *Sabotage Door*, and *Disable Module*) related to the scenarios identified. Similarly, some of these goals have also been decomposed into several sub-goals to denote how to achieve each parent goal. Roles are identified by clustering goals, and linking perceptions and actions (see details in Table 1).

3.2 Architectural Design Phase

Usually, applications need also reading and/or writing data to achieve the functionality related with some roles. This information is specified during the architectural design phase in the *data coupling diagram* (see Fig. 2a). A data is information managed by agents or beliefs representing agent knowledge about the environment or itself. For example, a data (*BDD*) should be available to validate every ticket introduced. Furthermore, there are data to count how many people cross each module (*Traffic*), to have control on the enclosure capacity level (*OccupationLevel*) and determine if a user goes in or goes out the enclosure (*ModuleConfiguration*). Therefore, notice that data are able to have any granularity.

Once the functionality of the roles has been described through their relations with other entities (percepts, actions, goals and data), the guidelines offered by Prometheus to decide the types of agents that are included in the system are applied during the second phase namely architectural design. In general terms, these guidelines consist of grouping roles to obtain the system. The cohesion and coupling criteria are used to decide which the best groupings are to ease their maintenance. These two concepts are essential to obtain a suitable software development that is distinguished by its maximum cohesion and minimum coupling. In the proposed case study, to achieve the system requirements, an agent is introduced for each role identified (see Fig. 2b).

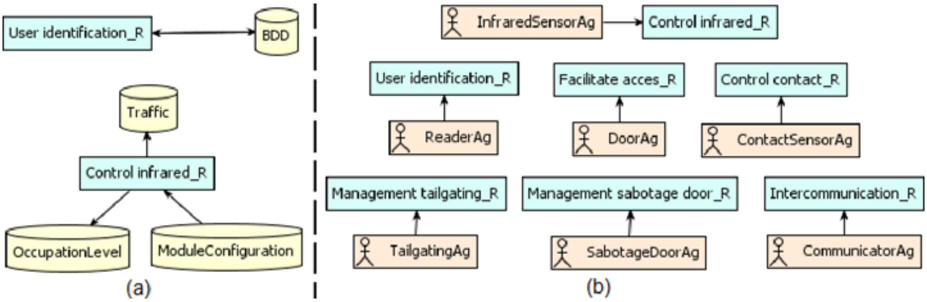


Fig. 2. (a) Data coupling diagram; (b) Agent-Role grouping diagram

Another relevant task performed in architectural design phase is to define the agent conversations (Interaction Protocols, IP) in order to describe what should happen to realize the specified goals and scenarios. Interaction Protocols are a graphical representation that shows (i) interactions among agents, and (ii) interactions among agents and the environment. It should be highlighted that percepts are originated by actors that communicate with agents, whereas actions describe a communication of agents with actors. For example, Fig. 3a details the *Tailgating_IP* interaction protocol internal structure, a sub-protocol of *Access_IP*. As can be noticed, it involves two agents and two actors (identified by the dotted squares in the diagram) to detect tailgating situations. This situation happens when the door remains open after an authorized person has passed through it and the related infrared sensor has been activated once. From this moment, whenever the *InfraredSensorAg* agent perceives that the infrared sensor beam has been broken again, a new crossing through the module is counted (*IncreaseTrafficCounter_a*), the number of people inside the enclosure is updated (*IncreaseOccupationCounter_a*), and the detection is communicated to the *TailgatingAg* agent sending a *BrokenBeam* message. Finally, the *TailgatingAg* agent executes the *NotifyTailgating_a* action in order to notify the *Central_A* actor an unauthorized access. This checking is carried out until the *ContactSensorAg* agent notifies the *TailgatingAg* agent that the door is closed thanks to the *Access_IP* protocol. Tailgating detection is carried out in the same way for entrance and exit alternatives.

Fig. 3b shows a fragment of *system overview diagram* for the proposed system design that includes the Tailgating Interaction Protocol. It can be observed that once roles have been grouped into agents, the information about percepts, actions, and data related to roles is automatically propagated and linked with the agents in the system overview diagram. Finally, the *agent acquaintance diagram* that contains the communication links between agents is automatically generated using the messages of information included in the interaction protocols. In short, every agent that perceives information captured by a device uses an intermediate agent (*CommunicatorAg*) to establish its communication. On the other hand, the agents responsible for detecting anomalous situations (*TailgatingAg* and *SabotageDoorAg*) can benefit from the knowledge offered by agents related

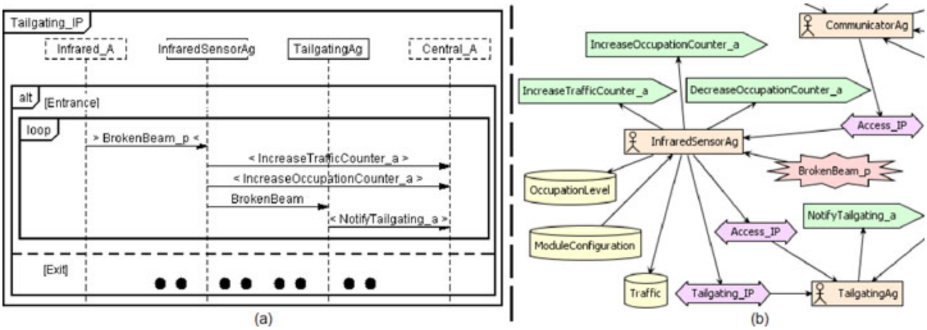


Fig. 3. (a) Tailgating interaction protocol; (b) Fragment of system overview diagram

to the contact and infrared sensors (*InfrarredSensorAg* and *ContactSensorAg*) to carry out their tasks.

3.3 Detailed Design Phase

Now, we are in front of one of the major aspects and contributions of this proposal. The system overview diagram previously obtained (see Fig. 2b) describes the overall system architecture, that is, the agents identified, the perceptions received, the actions to be executed, the interaction protocols among agents, and the data read and/or written by agents. These entities are described using Prometheus concepts. However, the *VigilAgent* models of the detailed design phase are developed using INGENIAS concepts, which are different from the previous ones. Therefore, a transformation from Prometheus models to INGENIAS models must be carried out in order to perform the next modeling phase.

We have developed four conceptual mappings [2] to transform the structures that involve percepts, actions, messages and data related to agents. These mappings have been inferred considering both the definition of these concepts, and how each Prometheus structure can be modeled using an INGENIAS equivalent structure. For example, a *percept* is a piece of information from the environment received by means of a sensor. Percepts are sent by *actors* (Actor → Percept) by means of *interaction protocols*, these percepts are received by agents (Percept → Agent). The relations among actors, percepts and agents (Actor → Percept → Agent) are described in the *system overview diagram*. In INGENIAS, all the software and hardware that interacts with the system and cannot be designed as an agent is considered an *application*; and every agent that perceives changes in the environment must be in the *environment model* associated to an application. Therefore, as Fig. 4 shows (arrow 1), the percepts of a Prometheus agent can be triggered in INGENIAS by specifying a collection of *operations* in an *application*. A Prometheus percept has a field, *Information carried*, to specify the information it carries. As Fig. 4 depicts (arrow 2), in INGENIAS this information is described in a type of event named *ApplicationEventSlots* that is associated to the *EPerceives* relation established between the agent and the

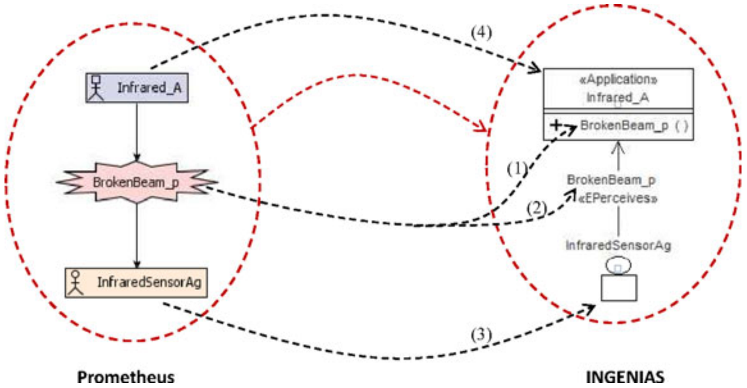


Fig. 4. Example of mapping information related with Prometheus percepts into INGENIAS

corresponding application. Notice that the Prometheus agent and actor concepts have been directly mapped to INGENIAS agent and application concepts (see arrows 3 and 4, respectively). Finally, let us highlight that action, message and data concepts in Prometheus are equivalents to operation defined in the applications, interaction unit and application concepts, respectively, in INGENIAS. The conceptual mappings have been automated specifying a Model-To-Model (M2M) transformation using the Query/View/Transformation (QVT) language.

Once the transformation has been performed, three activities should be carried out to complete the modeling. Firstly, it is necessary to identify the tasks performed by each agent by examining the *interaction model* and the initial *environment model* which are automatically generated from the Prometheus model applying the transformation. Specifically, an agent should perform a task for every received perception or message. After that, a *state diagram* describing the behavior of each agent is specified using information about tasks, received percepts and messages. Secondly, the *Applications*, obtained after once the transformation is executed, only include methods to send perceptions to the agents and to specify actions to be executed by the agents. Therefore, these applications should be refined to include new methods depending on the specific needs of the system being developed. For example, in the proposed case study, the *CheckUser* method was added to the *BDD* application to get information stored in a database, and the *GetId* method was added in *Reader_A* application to get its identifier. Finally, a model to describe the *application deployment* is specified attaching numbers in notes to represent the number of instances that there are in execution for each type of entity.

3.4 Implementation and Deployment Phases

Our approach considers that supporting tool for INGENIAS, Ingenias Development Kit (IDK) [7], is an exceptional agent tool to develop a Model-To-Text (M2T) transformation for generating code for any target language chosen, this is ICARO-T in *VigilAgent*, as it provides the necessary functionalities for developing new modules capable to carry out this task. These modules are developed following a general process based on both the definition of specific templates for each target platform, and procedures to extract information from INGENIAS models. In the literature there is no proposal for mapping INGENIAS models to ICARO-T code. So, our contribution to solve this gap has been to identify them and to develop modules to automatically carry out the Model-to-Text transformation from INGENIAS to ICARO-T.

The process for using the IDK modules in order to automatically generate ICARO-T code of the system being developed that can be updated with new code introduced by the user when necessary is carried out as follows. (1) The *INGENIAS ICARO-T Framework generator* module (*IIF*) is used to automatically generate code for the detailed design specification. IIF generates several XML files that describe the behavior of each agent, java classes for each agent and application, and the XML file describing the application deployment. (2) The developer manually inserts code in the protected regions of the generated java classes and implements those new classes he/she needs. (3) The developer uses the *ICAROTCodeUploader* module in order to update the model with the modifications introduced in the protected regions. It allows the developers to always keep the model and the source code synchronized, so that those changes introduced in the source code are kept when code is regenerated again from the model. Finally, the script file generated by IIF module is executed by the deployment manager to launch the developed application.

4 Conclusions

The lack of consensus about which the best methodology to develop agent-oriented applications is has driven the main idea presented in this paper: not to propose a new methodology from scratch but to combine several methodologies, *VigilAgent*, to take advantage of each one of the analyzed methodologies. The learning curve of *VigilAgent* can be steep at first because users must get used to different terms that have the same meaning depending on the technology used in each phase (Prometheus and INGENIAS for modeling, and ICARO-T for implementation). However, this disadvantage is overcome thanks to the two transformations that are executed automatically. First, a Model-to-Model transformation executed by means of Medini automatically transforms Prometheus structures into their equivalent INGENIAS structures. Second, the Model-to-Text transformation tool transforms INGENIAS models into code for the ICARO-T framework.

It is worth pointing out that the time spent learning how to develop and implement the INGENIAS ICARO-T Framework generator and the ICAROT-CodeUploader modules was two months and fifteen days. This effort is rewarded as new applications are modeled and implemented because (i) the quality of the developed system is improved as the automatically generated code does "not" have bugs; (ii) productivity is improved as the time necessary for coding is reduced because the developer does not need to learn the structure, location and naming rules for ICARO-T applications files; and (iii) more efforts can be devoted to solve errors during early phases of the life cycle, avoiding in this way the "snow ball" effect. Our future work consists in going on to apply *VigilAgent* to face new case studies.

Acknowledgements

This work was partially supported by the Spanish Ministerio de Ciencia e Innovación under projects TIN2010-20845-C03-01 and CENIT A-78423480, and by the Spanish Junta de Comunidades de Castilla-La Mancha under projects PII2I09-0069-0994 and PEII09-0054-9581.

References

1. Bordini, R.H., Dastani, M., Dix, J., El Fallah Seghrouchni, A.: Multi-Agent Programming: Languages, Platforms and Applications. Springer, Heidelberg (2005)
2. Gascueña, J.M., Fernández-Caballero, A.: Prometheus and INGENIAS agent methodologies: A complementary approach. In: Luck, M., Gomez-Sanz, J.J. (eds.) AOSE 2008. LNCS, vol. 5386, pp. 131–144. Springer, Heidelberg (2009)
3. Gascueña, J.M., Fernández-Caballero, A., Garijo, F.J.: Using ICARO-T framework for reactive agent-based mobile robots. In: Advances in Practical Applications of Agents and Multiagent Systems. Advances in Soft Computing, vol. 70, pp. 91–101. Springer, Heidelberg (2010)
4. Gascueña, J.M., Fernández-Caballero, A.: On the use of agent technology in intelligent, multi-sensory and distributed surveillance. The Knowledge Engineering Review (2011) (in press)
5. Padgham, L., Winikoff, M.: Developing intelligent agents systems: A practical guide. John Wiley and Sons, Chichester (2004)
6. Patricio, M.A., Castanedo, F., Berlanga, A., Pérez, O., García, J., Molina, J.M.: Computational intelligence in visual sensor networks: Improving video processing systems. SCI, vol. 96, pp. 351–377. Springer, Heidelberg (2008)
7. Pavón, J., Gómez-Sanz, J.J., Fuentes, R.: The INGENIAS methodology and tools. In: Agent-Oriented Methodologies, pp. 236–276. Idea Group Publishing, USA (2005)
8. Pavón, J., Gómez-Sanz, J.J., Fernández-Caballero, A., Valencia-Jiménez, J.J.: Development of intelligent multi-sensor surveillance systems with agents. Robotics and Autonomous Systems 55(12), 892–903 (2007)
9. Räty, T.D.: Survey on contemporary remote surveillance systems for public safety. IEEE Transactions on systems, man, and cybernetics - Part C: Applications and reviews 40(5), 493–515 (2010)
10. Rivas-Casado, A., Martínez-Tomás, R., Fernández-Caballero, A.: Multiagent system for knowledge-based event recognition and composition. Expert Systems: The Journal of Knowledge Engineering (2011) (in press)

Clustering of Trajectories in Video Surveillance Using Growing Neural Gas

Javier Acevedo-Rodríguez[1], Saturnino Maldonado-Bascón[1],
Roberto López-Sastre[1], Pedro Gil-Jiménez[1],
and Antonio Fernández-Caballero[2,3]

[1] University of Alcala, Teoría de la señal y Comunicaciones, Alcalá de Henares, Spain
http://agamenon.tsc.uah.es/Investigacion/gram/index.html

[2] Instituto de Investigación en Informática de Albacete (I3A), Universidad de Castilla-La Mancha, 02071 Albacete, Spain

[3] Departamento de Sistemas Informáticos, Universidad de Castilla-La Mancha, 02071 Albacete, Spain

Abstract. One of the more important issues in intelligent video surveillance systems is the ability to handle events from the motion of objects. Thus, the classification of the trajectory of an object of interest in a scene can give important information to higher levels of recognition. In this context, it is crucial to know what trajectories are commonly given in a model in order to detect suspect ones. This implies the study of a set of trajectories and grouping them into different categories. In this paper, we propose to adapt a bioinspired clustering algorithm, growing neural gas, that has been tested in other fields with high level of success due to its nice properties of being unnecessary to know a priori the number of clusters, robustness and that it can be adapted to different distributions. Due to the fact that human perception is based on atomic events, a segmentation of the trajectories is proposed. Finally, the obtained prototype sub-trajectories are grouped according to the sequence of the observed data to feed the model.

Keywords: Trajectory clustering, growing neural gas, high-level video surveillance.

1 Introduction

The research on intelligent video surveillance systems has been intensive for the last years with emphasis on obtaining high-level information for interpreting a scene in order to give some kind of alarms when an event detection has been triggered. One of the issues that has gained attention is to automatically recognize behaviors, where the movement followed by the objects of interest can give rich information about specific or suspicious behaviors. Trajectory classification is then placed between low level stages such as segmentation, object recognition or tracking and high level interpretation of the scene. The trajectories followed by objects of interest in the image is a valuable source of information to automatically detect these suspicious behaviors or generate some kind of alarms [13].

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 461–470, 2011.

In order to define the model of the surveillance system it is of high interest to associate objects with common observed trajectories. As the number of trajectories grows it becomes impractical to group and identify them manually, so it is necessary to use a clustering algorithm that can also provide prototypes of trajectories and detect outliers.

Trajectory matching involves the measure of the similarity respect to some kind of distance metric between two or more trajectories. The most trivial distance measure is the Euclidean one, but it should be noted that two different trajectories can have different number of points. In [9] trajectories are re-sampled so that all of them have the same number of points, allowing the use of the Euclidean distance as a metric of similarity. Nevertheless, this measure is suboptimal when comparing sub-trajectories obtained as a result of some kind of occlusion in the tracking stage. This fact makes necessary the research on other measures that do not require the two trajectories to have the same number of points. One of the measures in this second group, widely used in trajectory comparison, is Dynamic Time Warping (DTW) [19], [18]. Another measure that do not need the same number of points is the Longest Common SubSequence (LCSS) used in [5] or [7]. The modified Hausdorff distance introduced in [2] adapts the Hausdorff distance to compare trajectories keeping the order of the points in the sequences. In [14] the distance used was called Trajectory Directional Histogram and it is based on the histogram of the angles obtained from the sequence.

Previous works have explored clustering methods in order to group trajectories. The most extended is K-means and its soft version fuzzy K-means, used in [12] or [9]. Agglomerative techniques have also been used as described in [1] or [5]. A common problem with these approaches is that it is necessary to know a priori the number of centroids or classes of the problem and make some assumptions about the distribution of the data. Spectral clustering is another method found in the literature, that do not make any assumption on the distribution of data points and approximates an optimal graph partition [16]. Its use in trajectory clustering is described in [3]. However, this last approach does not provide prototype trajectories, which can be one of the targets if we are interested in a model of the video scene. A deeper review of the state of the art in trajectory classification and clustering can be found in [15]. More recently, several works indicate that the problem of trajectory classification and clustering can be easily solved by dividing it into sub-trajectories. This solution is based on the way humans recognize trajectories by dividing them into atomic units of actions that are of substantial value for perception [4], [17].

Growing neural gas [8] (GNG) is a bioinspired algorithm for finding optimal representation of feature vectors. Its name comes from the behaviour of the vectors during the adaptation process that distributes them like a gas in space. It is based on growing self organizing maps and has been used for clustering analysis [6]. The advantage as a clustering algorithm is that, with enough training time, it can adapt to clusters with very different nature, keeping more neurons in those clusters that are more complex to be represented. This paper proposes to adapt this algorithm to the problem of trajectory clustering using segments of the trajectories observed in a video scene. The obtained weights indicate prototype sub-trajectories that are useful to generate prototype trajectories to feed the model. To this end, an algorithm that studies sequences of prototype trajectories was designed. This algorithm studies the observed trajectories and associates them to one of the clusters provided by the GNG algorithm.

2 Proposed Algorithm

2.1 Trajectory Segmentation

Let's suppose that a trajectory T_i is defined by a set of n_i 2D-points, corresponding to consecutive positions of a tracked object of interest, observed at equally spaced intervals of time, $T_i = \{(x_1, y_1), (x_2, y_2), ..., (x_{n_i}, y_{n_i})\}$. As mentioned, the number of points n_i of each trajectory can change from one trajectory to other, and depends on several factors such as the path followed by the object, the velocity or a truncation in the trajectory due to some occlusions. As it was described in the preivous section, it is more complex to define a distance measure when the whole trajectory is considered. In this paper we propose to divide each trajectory in linear segments by using a fast realization of the Douglas-Peucker (DP) algorithm [11]. This algorithm was already used in trajectory compression [10] but in this case, we propose to use these segments as the inputs of the GNG network. As it is shown in Figure (1) after the DP algorithm has been applied to trajectories, these are transformed into $T_i \rightarrow S_i = \boldsymbol{s}_1^i, \boldsymbol{s}_2^i, .., \boldsymbol{s}_d^i$, being $d_i << n_i$ the number of segments extracted, and each of the elements $s_j^i \in \mathbb{R}^4$ is composed by the coordinates of the starting point (x_s, y_s) and the end point (x_f, y_f) concatenated. Segments are then treated as vectors, all of them with the same dimension, and then the clustering algorithm is expected to find prototype segments.

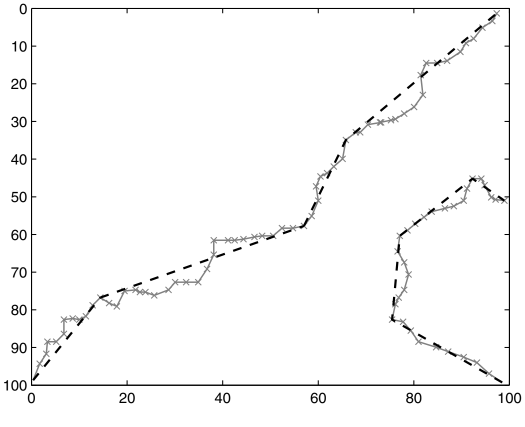


Fig. 1. Douglas-Peucker algorithm applied to extract linear segments from trajectories

The only parameter to be adjusted on this stage is τ defined as:

$$\tau = \max_i \left(\min_j d\left((x_i, y_i), \boldsymbol{s}_j\right) \right) . \tag{1}$$

where $d\left((x_i, y_i), \boldsymbol{s}_j\right)$ is the distance between point (x_i, y_i) of the original trajectory and the segment $\boldsymbol{s}_j$ of the linearized trajectory.

2.2 Growing Neural Gas

Once we have extracted all the segments from the observed set of trajectories, the GNG algorithm is used to find clusters of sub-trajectories. The data used in the training process is the whole set of segments, from all the trajectories, found in the previous stage. This training set is defined as $\boldsymbol{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ...\boldsymbol{s}_m\}$, where m is the total number of segments. It is important to highlight that, in order to find the clusters, the information about what trajectory is producing the segment $\boldsymbol{s}_i$ is not present. The procedure can be summarized in the following steps:

1. Define a maximum number of iterations, *mxiter*, a maximum number of nodes *mxnodes*, the actual number of nodes $n = 0$, a matrix of weights $\boldsymbol{\Omega} = \emptyset$, a vector of errors, $\mathrm{E} = 0$, a matrix of connections, $\mathrm{C} = 0$, and set the actual iteration $iter = 1$.
2. Randomly select two segments from the training set, $(\boldsymbol{s}_a, \boldsymbol{s}_b)$ and assign them as the weights of the two first nodes:

$$\begin{array}{l} \boldsymbol{\omega_1} \leftarrow \boldsymbol{s}_a \\ \boldsymbol{\omega_2} \leftarrow \boldsymbol{s}_b \\ \boldsymbol{\Omega} = \{\boldsymbol{\omega_1}, \boldsymbol{\omega_2}\} \end{array} . \tag{2}$$

   Set a connection $C(1, 2) = C(2, 1) = 1$ and make $n = 2$.
3. Randomly select a segment,$\boldsymbol{s}_i$ from the training set and search the nearest weight, $\boldsymbol{\omega_f}$ and the second one $\boldsymbol{\omega_s}$. Connect nodes f and s, $C(f, s) = C(s, f) = 1$. Increment the error associated to node f by:

$$E_f = E_f + \|\boldsymbol{s}_i - \omega_f\|^2 \tag{3}$$

   Update the reference vectors of the winner and its direct topological neighbors by fractions ε_a and ε_n, respectively, of the total distance to the input vector:

$$\varDelta\boldsymbol{\omega}_f = \varepsilon_a\left(\boldsymbol{s}_i - \omega_f\right); \;\; \varDelta\boldsymbol{\omega}_j = \varepsilon_b\left(\boldsymbol{s}_i - \omega_j\right) \; j \in \boldsymbol{N}_f \tag{4}$$

   where $\boldsymbol{N}_f$ is the set of direct topological neighbors of node f.
4. Increment the connection value of all neighbors of f, $C(f, j) = C(f, j) + 1, C(j, f) = C(j, f) + 1$. Remove all the connections, $C(f, h) = C(h, f) = 0$, greater than a predefined value α. If in this step a node is found to be isolated, it is removed from all associated vectors and matrices and the number of active nodes is decreased. Increment the value of *iter*.

5. If $mod(iter/\lambda) = 0$ and $n < mxnodes$, where λ is a predefined value, a new node r is inserted. To this end, the node with maximum error is found,

$$q = \max_{i\ \boldsymbol{\omega}_i \in \Omega} E(i)\ . \tag{5}$$

and also the node with maximum error among the neighbors of q,

$$l = \max_{i\ \boldsymbol{\omega}_i \in \boldsymbol{N}_q} E(i)\ . \tag{6}$$

The new node is inserted by breaking the connection between nodes q and l, $C(q, l) = C(l, q) = 0$ and adding the new connection $C(r, q) = C(r, q) = C(l, r) = C(r, l) = 1$. The weight of the new node is:

$$\begin{array}{l} \boldsymbol{\omega}_r = (\boldsymbol{\omega}_l + \boldsymbol{\omega}_q)/2 \\ \Omega = \Omega \bigcup \boldsymbol{\omega}_r \\ E(r) = (E_l + E_q)/2 \end{array} \tag{7}$$

Increase the number n of active nodes.
6. Decrease the error in all the nodes $\Delta E(i) = -\beta E(i)$ where β is a predefined constant. If $iter < mxiter$ repeat the process from point 3. Else stop the algorithm and return the matrix $\boldsymbol{\Omega}$.
7. Find those nodes that are enough close each other, $\|\boldsymbol{\omega}_i - \boldsymbol{\omega}_j\| < \mu$, where μ is a predefined constant and group them into a unique node.

2.3 Prototype Trajectories

The GNG algorithm provides a set of n nodes, each one representing a subtrajectory prototype. In this stage the algorithm searches for common sequences of sub-trajectories in order to build a set of prototype trajectories. These common sequences are extracted from the set of observed trajectories. As it has been highlighted in the previous section, there was a lack of information in the GNG algorithm about what trajectory produce each segment. Now, we have recovered the information obtained in the first step, so instead of considering initial trajectories the sequences of linear segments are studied. Each segment is associated to its nearest node:

$$\begin{array}{l} T_i \rightarrow S_i = \boldsymbol{s}_1^i, \boldsymbol{s}_2^i, .., \boldsymbol{s}_d^i \\ \phi_j^i = \max_z \|\boldsymbol{s}_j^i - \boldsymbol{\omega}_z\|\ j = 1, 2, ..d_i\ z = 1, 2, .., n \end{array} \tag{8}$$

From this assigment, we have mapped each trajectory into a set of indexes associated to prototype sub-trajectories, $\boldsymbol{\Phi}_i = \{\phi_1, \phi_2, ...\phi_{d_i}\}$. A search is now made to obtain those $\boldsymbol{\Phi}_i$ that are repeated more than a given number of times. In our work we have considered a sequence of sub-trajectories to be a prototype if it is repeated more than four times.

3 Results and Discussion

3.1 Clustering Performance

First of all we would need to verify if the GNG method fulfills the requirements of grouping sub-trajectories. To this end we have build an artificial dataset of segments shown in Figure (2) and each one was assigned to a group. We define the accuracy of a clustering algorithm as:

$$\rho = \frac{1}{n} \sum_{i=1}^{k} \sum_{j=1}^{n} \frac{I_i(j)}{1 + O_i(j)} \ . \tag{9}$$

where n is the number of clusters, in our case nodes, k is the number of classes, $I_i(j)$ is the number of samples of class i that are associated to cluster j and $O_i(j)$ is the number of elements that do not belong to class i but are present in cluster j. This criteria is maximized in the case we have the minimal number of clusters that only contain samples of one class. One sample is said to belong to a cluster j if the euclidean distance with this cluster is the minimal among all the distances with clusters and its value is not higher than a threshold. This definition is necessary in the presence of outliers, such as those that can be observed in our example. In our case we have set the threshold value of 30.

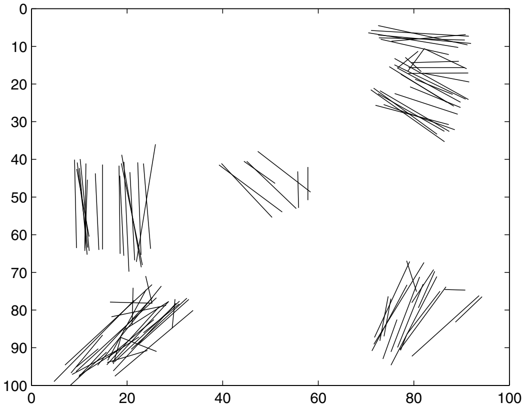


Fig. 2. Artificial problem with different segments distributed in groups and noise

We want to check the influence of two important parameters *maxiter* and λ. Results of the proposed example are shown in Table (1), where the algorithm was executed without the final cluster aggregation proposed in the last step of the algorithm.

From these results, we can extract important conclusions that must be kept in mind when the GNG algorithm is executed. When the number of iterations is very high and λ is very low the algorithm degenerates to every single sample is

Table 1. Cluster Precision for different values of *maxiter* and λ

λ	*maxiter*				
	3000	25000	50000	100000	300000
500	4.12	6.26	7.45	1	1
2000	NA	8.42	13.33	5.26	1
6000	NA	5.36	8.25	11.42	6.52
10000	NA	4.2	5.36	6.42	7.21

a cluster. When the number of iterations is very low and λ is also low there are not enough nodes and their weights are not properly adapted. The extreme case would happen when we have an only cluster. We can check that selecting these parameters produce an adequate number of clusters. In Figure (3) it is shown the prototype segments found with $maxiter = 50000$ and $\lambda = 2000$.

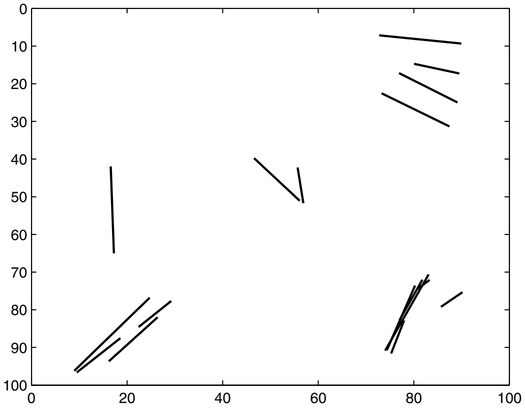


Fig. 3. Nodes found by the GNG method with $maxiter = 50000$ and $\lambda = 2000$

3.2 Trajectories in a Scene

In the previous section it was demonstrated how the GNG algorithm performs appropriately for the problem of noisy segment clustering. In this section the whole algorithm is tested within an image of the entrance lobby of the INRIA Labs at Grenoble, filmed for the CAVIAR project with a wide angle camera. It is necessary to say that, although the image come from real video scene application, the trajectories we have worked on are synthetic trajectories. This was needed to verify the behaviour of the method with complex trajectories and it is usually done in most of the works for trajectory clustering. In Figure (4) some of the synthetic trajectories are shown. It can be noticed how some of them are only parts of completed trajectories, included to mimic the effects of occlusions and problems with tracking algorithms.

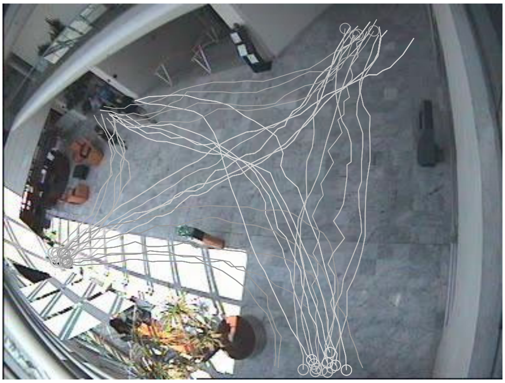


Fig. 4. Synthetic observed trajectories in the INRIA scene

In Figure (5) the prototype trajectories found by the algorithm are shown. It should be noted that several trajectories share some of the segments, although in the image they has been represented only in one color. This is logical because we can see in Figure (4) that many trajectories start from narrow paths and disperse later. The parameters used to obtain these results were $\tau = 5$ for the initial DP algorithm to extract segments; the GNG algorithm was tuned according to results obtained in the previous section with $maxiter = 50000$, $\lambda = 2000$, $\varepsilon_a = 0.07$, $\varepsilon_b = 0.008$, $\beta = 0.0005$ and $\alpha = 40$. The μ constant defined to group segments that are closer enough each other was fixed to $\mu = 1000$.

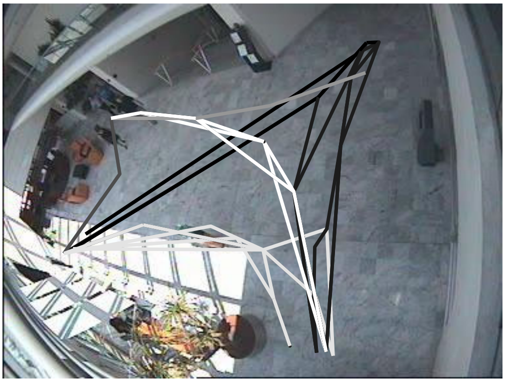


Fig. 5. Prototype trajectories found with the proposed method

As it can be noticed, the algorithm finds prototype trajectories that can feed a model in order to detect abnormal behaviour when a trajectory is segmented and their sequence of segments is far away from any trajectory found by the method. It should be said that partial trajectories have also been used to build the model, and their segments are used to train the GNG. Opposite to other trajectory clustering methods described in the literature, we do not need to select only complete trajectories and the information of these pieces of trajectories are exploited but as these incomplete trajectories are not frequently repeated they do not build a prototype.

4 Conclusions

In this paper we have introduced a new method to obtain prototype trajectories. Based on recent works that suggest that the problem of trajectory matching can be better solved if pieces of trajectories are considered, we start the problem by dividing observed trajectories into linear segments. The GNG algorithm is used then to find sub-trajectory prototypes and demonstrates that has a good behaviour due to its robustness against outliers and that a number of clusters is not needed a priori. These prototypes are used to build sequences of common subtrajectories, very useful to detect events in higher levels. Very promising results have been obtained using synthetic observed trajectories in real scenes.

Future work will be focused on working with real observed trajectories and feed back the tracking algorithm with the information provided by the trajectory identification algorithm, once that the prototypes have been learned.

Acknowledgement

This work was supported by the Spanish Ministry for Science and Innovation under Projects TIN2010-20845-C03-03, TIN2010-20845-C03-01, Plan Avanza TSI-020302-2009-59 and by the Madrid Government under project CCG10-UAH/TIC-5965.

References

1. Antonini, G., Thiran, J.: Counting pedestrians in video sequences using trajectory clustering. IEEE Transactions on Circuits and Systems for Video Technology 16(8), 1008–1020 (2006)
2. Atev, S., Masoud, O., Papanikolopoulos, N.: Learning traffic patterns at intersections by spectral clustering of motion trajectories. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4851–4856 (2006)
3. Atev, S., Miller, G., Papanikolopoulos, N.: Clustering of vehicle trajectories. IEEE Transactions on Intelligent Transportation Systems 11(3), 647–657 (2010)
4. Bashir, F.I., Khokhar, A., Schonfeld, D.: Object trajectory-based activity classification and recognition using hidden markov models. IEEE Transactions on Image Processing 16(7), 1912–1919 (2007)
5. Buzan, D., Sclaroff, S., Kollios, G.: Extraction and clustering of motion trajectories in video. In: International Conference on Pattern Recognition, pp. 521–524 (2004)
6. Canales, F., Chacón, M.: Modification of the growing neural gas algorithm for cluster analysis. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 684–693. Springer, Heidelberg (2007)

7. Cheriyadat, A., Radke, R.: Automatically determining dominant motions in crowded scenes by clustering partial feature trajectories. In: Proceedings of the First ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC 2007 (2007)
8. Fritzke, B.: A growing neural gas network learns topologies. In: Advances in Neural Information Processing Systems, vol. 7, pp. 625–632 (1995)
9. Fu, Z., Hu, W., Tan, T.: Similarity based vehicle trajectory clustering and anomaly detection. In: IEEE International Conference on Image Processing, ICIP 2005, vol. 2, pp. 602–605 (2005)
10. Gudmundsson, J., Katajainen, J., Merrick, D., Ong, C., Wolle, T.: Compressing spatio-temporal trajectories. In: Tokuyama, T. (ed.) ISAAC 2007. LNCS, vol. 4835, pp. 763–775. Springer, Heidelberg (2007)
11. Hershberger, J., Snoeyink, J.: Speeding up the douglas-peucker line-simplification algorithm. In: Proc. 5th Intl. Symp. on Spatial Data Handling, pp. 134–143 (1992)
12. Hirasawa, N.S.K., Tanaka, K., Kobayashi, Y., Sato, Y., Fujino, Y.: Learning motion patterns and anomaly detection by human trajectory analysis. In: IEEE International Conference on Systems, Man and Cybernetics, ISIC 2007, pp. 498–503 (2007)
13. Jung, C., Hennemann, L., Musse, S.: Event detection using trajectory clustering and 4-d histograms. IEEE Transactions on Circuits and Systems for Video Technology 18(11), 1565–1575 (2008)
14. Li, X., Hu, W., Hu, W.: A coarse-to-fine strategy for vehicle motion trajectory clustering. In: 18th International Conference on Pattern Recognition, ICPR 2006, vol. 1, pp. 591–594 (2006)
15. Morris, B., Trivedi, M.: A survey of vision-based trajectory learning and analysis for surveillance. IEEE Transactions on Circuits and Systems for Video Technology 18(8), 1114–1127 (2008)
16. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14, pp. 849–856. MIT Press, Cambridge (2001)
17. Piciarelli, C., Foresti, G.: On-line trajectory clustering for anomalous events detection. Pattern Recognition Letters 27(15), 1835–1842 (2006)
18. Pop, I., Scuturici, M., Miguet, S.: Incremental trajectory aggregation in video sequences. In: 19th International Conference on Pattern Recognition, ICPR 2008, pp. 1–4 (2008)
19. Yanagisawa, Y., Satoh, T.: Clustering multidimensional trajectories based on shape and velocity. In: Proceedings of 22nd International Conference on Data Engineering Workshops, p. 12 (2006)

Human Action Recognition Based on Tracking Features

Javier Hernández, Antonio S. Montemayor,
Juan José Pantrigo, and Ángel Sánchez

Departamento de Ciencias de la Computación
Universidad Rey Juan Carlos, C/Tulipán, s/n,
28933 Móstoles, Madrid, Spain
{javier.hernandez,antonio.sanz,juanjose.pantrigo,angel.sanchez}@urjc.es

Abstract. Visual recognition of human actions in image sequences is an active field of research. However, most recent published methods use complex models and heuristics of the human body as well as to classify their actions. Our approach follows a different strategy. It is based on simple feature extraction from descriptors obtained from a visual tracking system. The tracking system is able to bring some useful information like position and size of the subject at every time step of a sequence, and in this paper we show that, the evolution of some of these features is enough to classify an action in most of the cases.

1 Introduction

Human action recognition aims to understand patterns of human movement from image sequences and classify those actions into known categories. This is a relevant problem in computer vision since it has applications in video surveillance and monitoring human-computer interactions, augmented reality, and so on [1].

Human actions consist of spatial-temporal patterns that are generated by a complex and time varying non-linear dynamic system. A complete description of the system requires enumeration of all the variables, their interdependencies, equations controlling their evolution and a set of boundary conditions to be satisfied by the system [9]. Usually, the processing of this description needs too many computational resources becoming intractable in real time for most of the cases.

An standard approach for human action recognition is to extract a set of features from each image sequence and use it to train classifiers to perform recognition. Using those properties a system can classify or approximate a model and use this model to classify. Approaches can be grouped depending on the image properties such as motion-based, shape-based, gradient-based, etc. [3].

Several features have been proposed in the literature. In Wang and Suter [1] some features extracted from human silhouettes or from their distance transform are classified using three different methods: Gaussian mixture models, matching based with the Hausdorf distance and continuous hidden Markov models. Zhou et al. [4] employed time-space human silhouettes that are transformed into low

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 471–480, 2011.

dimensional multivariate time series with Tensor Subspace Analysis in order to classify with a Gaussian process, Support Vector Machines (SVM), etc. Ali et al. [9] used a combination of chaotic invariants as feature vector that are classified using K-nearest neighbours. A keypose based approach where each action is modelled as series of synthetic poses and constraints on transitions between them is used in [5]. In [7] each action is represented as a unique curve in a 3D invariance-space and matched against candidate action volumes in a probabilistic framework. Finally optical flow volumetric features have also been studied in [8].

We propose the use of a visual tracking system in order to get the height and width as well as the position of a bounding box containing the subject in every time step of a sequence. The evolution in time of these bounding box properties are used as the input of a SVM classifier. The visual tracking task is performed by a variant of a particle filter [12] that includes a memetic algorithm [14] as a refinement stage. We called this method as Memetic Algorithm Particle Filter (MAPF). For a detailed description of MAPF please refer to [16]. The rest of the paper is organized as follows. Section 2 presents the MAPF as well as the PF and MA algorithms. Section 3 describes the feature extraction module. Section 4 presents the experimental results. Finally, section 5 summarizes the conclusions and future lines of research.

2 The MAPF Algorithm

2.1 Particle Filters for Visual Tracking

Particle Filters approximate theoretical distributions in the state-space by simulated random measures called particles [18]. The state-space model consists of two processes: (i) observation process in which observation vector is obtained, and (ii) a transition process. The goal is to estimate the new system state at each time step.

In the framework of Sequential Bayesian Modelling, the posterior *pdf* is estimated in two stages (i) evaluation in which the posterior is computed using the observation vector and (ii) propagation where the posterior is propagated at time step t.

A predefined system model is used to update the particle set. The problem lies in a state modelling where the dynamics equation describes the evolution of the object and the measurement equation links the observation with the state vector.

The aim is the recursive estimation of the posterior *pdf*, that constitutes a complete solution to the sequential estimation problem. This *pdf* is represented by a set of weighted particles. Each particle stores a system state at a given time and a quality measure, proportional to the probability of the state in represents.

The PF algorithm starts by initializing a population vector using a known *pdf*. The measurement vector at a given time step is obtained from the system, and particle weights are computed using a fitness function. The weights are normalized and a new particle set is selected. Taking into account that particles with larger weight values can be chosen several times, a diffusion stage is applied to avoid the loss of diversity. Finally, particle set for the next time step is predicted using the motion model. The pseudo-code of PF is detailed in [15].

In short, Particle Filters are algorithms that handle the evolution of particles. Particles are driven by the state model and are multiplied by their fitness values as determined by the *pdf* [18]. In visual tracking problems, this *pdf* represents the probability that the object is in a determined position.

2.2 Memetic Algorithms

The term memetic algorithm (MA) refers to a class of metaheuristics based on a population of agents [14]. This method has been successfully applied to a variety of optimization problems [17]. MA allows us to exploit all available knowledge about the problem under study. That is what makes MA different from other evolutionary methods. This philosophy is illustrated with the term '*meme*' , which denotes an analogy with the '*gene*' in the context of cultural evolution. The ideas ('*memes*') are propagated from brain to brain via the cultural processes in a way similar to how the gene pool ('*genes*') is propagated from body to body via reproduction processes.

The key idea of the MA is the combination of individual improvement procedures with cooperation and competition processes in a populational context. Algorithmically, MA maintains a population of agents during the whole optimization process. Agents are basically solutions of the problem. These agents are related to each other in a competition and cooperation context, which is organized in different generations. Each generation is a new update of the population, which is performed by recombining the features of some selected agents and replacing some agents with the new ones. The selection and replacement procedures are both competitive processes, while the combination stage is a cooperation process in which the selected agents generate new ones by applying reproductive operators (combination and mutation). Finally, the MA allows the application of local improvement procedures on some of the agents. The improvement procedure can be used at different stages of the optimization process, for example: as a mutation operator, only at the end of the process, etc.

2.3 The Hybrid Method: Memetic Algorithm Particle Filter

MAPF is the result of hybridize the memetic algorithm (MA) as optimization procedure and the particle filter (PF) as prediction procedure in two different stages:

- PF stage is focused on the temporal evolution of a representative set of solutions. A solution set, called SupportSet, of size N is propagated in time and updated to obtain a new set in every time step. The aim when using PF is tracking multiple hypotheses and using the knowledge about the system dynamics for future prediction.

– In the MA stage, solutions are combined and improved to obtain new better ones using the strategies of the Memetic Algorithm metaheuristic. This stage is a refinement step embedded in the PF framework and is based on a set of procedures extracted from MA general schema: selection, combination, mutation and improvement.

MAPF addresses the search to regions of the solution space in which it is highly probable to find new better solutions. MA stage performs a rational search beyond the simple stochastic procedure used by PF. On the other hand PF stages increase the performance of general optimization algorithms in dynamic problems by improving the quality of the diverse initial solution set. MA and PF are related in such a way that when MA improves its results, PF performance also improves, and vice versa.

MAPF can handle different size state-space representation in each stage according to the number of estimated features. In our case we track a single object and estimate its bounding box $[x, y, Lx, Ly]$ represented by position and size. It is possible to estimate an approximate position using the PF stage, so the PF stage determines the position $[x, y]$ resulting in a two dimensional problem. Memetic Algorithm refines this estimation considering the size of the target in this case the associate state-space of the MA is $[x, y, Lx, Ly]$. This is very useful to reduce the workload of the particle filter when using high dimensional state-spaces.

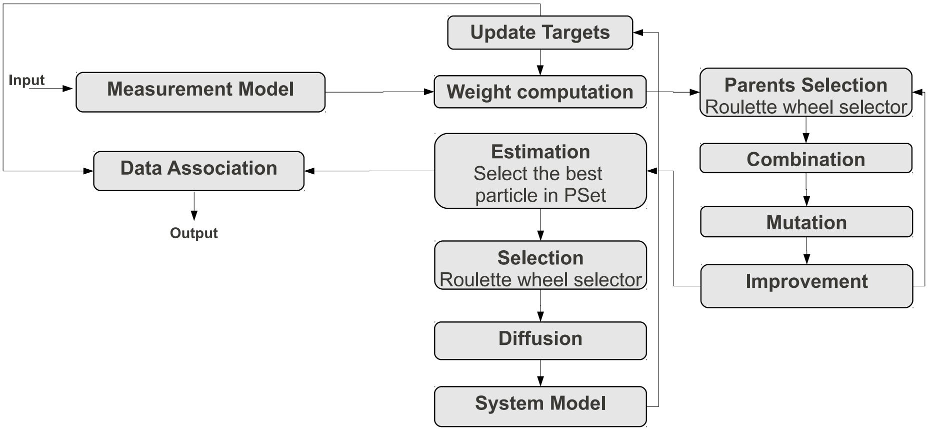


Fig. 1. Algorithmic details of the MAPF as the core of the tracking module

2.4 Structure of MAPF Output

The MAPF manages a population of solutions where each solution is stored in a particle. Each solution contains the set of variables which describe the system-state and its weight. As a result, the proposed state-space model for object tracking is 4 dimensional space. The structure which stores a solution is a state vector $s^t = [x^t, y^t, Lx^t, Ly^t]$, where (x^t, y^t) represents the position of the object o geometrical center in the global image frame at time t, and (Lx^t, Ly^t) is the height and width of a bounding box fitting the object o in the frame at time t.

3 Feature Extraction

An standard approach of a human action recognition system is to extract a set of features from each image sequence and use it to train a classifier that will eventually perform the recognition. Therefore, a key point in this approach is to select a robust feature set which allows to accurately discriminate the considered actions.

Feature selection is based on expert human knowledge. This model has a significant interest because it relies directly on an expression of facts stated in natural language. The model can be validated and allows to check the consistency of the set of rules. Furthermore, the model can be progressively completed by asking new questions in natural language.

Here is an example illustrating some knowledge used to select features:

- When someone is jumping in place there is no displacement
- The bend down action implies change in the height of the bounding box
- The jack action produces changes in the bounding box height and width
- While a subject walks the center of mass varies in a known direction

Considering the kind of rules stated above the idea is to find some features that can be extracted from the subject bounding box evolution through the time. The human knowledge will be then translated to rules involving these features. Variances, covariances, means, maximums and minimums of the variables that define bounding boxes are used to represent concepts as can be seen in Table 1. In order to test that those measures are discriminant enough a classification tree using fixed thresholds was built. The classification tree was tested against the database found in [11] and the classification was successful. There are measurable relations between these variables and the action being performed. An example of a person walking is described in Figure 2 where Lx shows a large variance while Ly remains almost constant.

Table 1. Some used variables and their meanings related to human knowledge statements about actions

Variable	Meaning
var(x)	displacement
var(y)	jump
var(Lx)	amplitude of the movement
var(Ly)	change of height
cov(Lx, Ly)	arm-leg coordination
cov(y, Lx)	arm-leg coordination

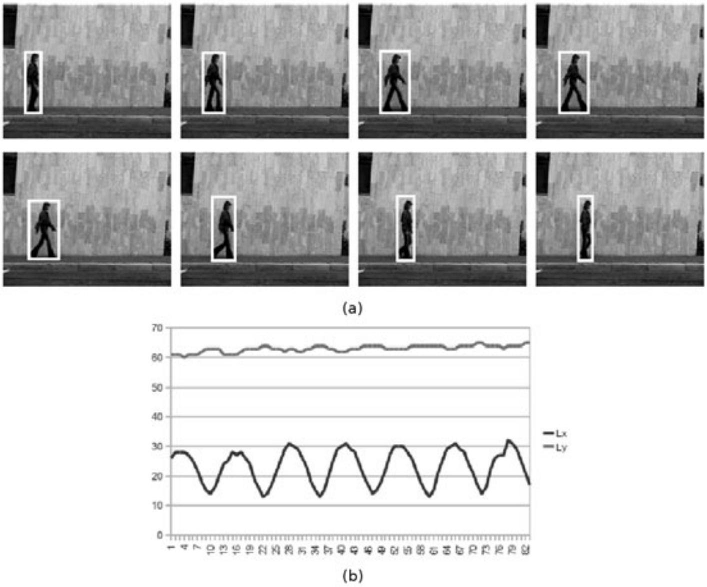


Fig. 2. (a) Evolution in time of the bounding box. (b) Variation of Lx and Ly associated to the bounding box.

4 Experimental Results

All the experiments were performed on a 3GHz Intel®Core$^{\mathrm{TM}}$2 Duo with 3GB RAM. The parameter configuration selected for the results achieved is summarized in Table 2.

Table 2. Parameter configuration

Parameter	description	value
N	Number of particles	30
I	Number of iterations in the MA stage	5
$[th_R, th_G, th_B]$	Measurement model thresholds	$[45, 45, 45]$
$[min_F, max_F]$	Diffusion range for $[x, y]$ in the system model and in the mutation method	$[-5, 5]$
$[min_G, max_G]$	Diffusion range for $[\dot{x}, \dot{y}]$ in the system model	$[-1, 1]$
$[\Delta x_o, \Delta y_o, \Delta Lx_o, \Delta Ly_o]$	Movement length values in the improvement method	$[2, 2, 1, 1]$
C	Penalty parameter for the SVM classifier	32768
γ	Kernel parameter for SVM	0.00012207031

The parameters used for the MAPF system are the same than in [16]. The parameters used for the SVM classifier were the result of exhaustive search over the parameter space using grid-search and cross-validation. Grid-search provides an exhaustive parameter search and the computational time required is low because there are only two parameters. Furthermore grid-search can be easily parallelized.

Experimental analysis is carried out on a dataset provided by [11]. This dataset consists of 9 subjects performing a set of 10 different actions (see Figure 3): *bending down, jumping jack, jumping, jumping in place, running, galloping sideways, hop-skip, walking, waving one hand and waving both hands.* Experiments were performed using 9 fold cross-validation over 90 samples: 8 subjects are used for training and the remaining one for testing; this procedure is repeated for every permutation and results are finally averaged.

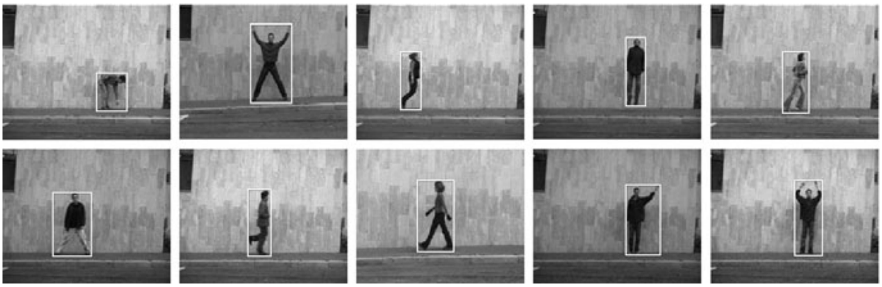


Fig. 3. Action database: examples of sequences corresponding to different types of actions

The tracking of the subjects was made using MAPF in order to achieve real time performance. MAPF is a very fast algorithm and is able to obtain high framerates even tracking a variable number of objects. In our case only one subject is being tracked. For this special case MAPF obtains 98.82 *fps*. Results for a variable number of objects can be found in [16].

The output of the MAPF tracking system is then processed to get the feature descriptors and they are finally sent to the classifier. SVM with a radial basis function kernel was used as classification method. The first performed step is data scaling, the main advantage of scaling is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges and some numerical difficulties present when using kernel because kernel values usually depend on the inner products of feature vectors. Scaling constants are calculated with training data and saved in order to use the same values with test data. In our case the range $[-1, +1]$ is used.

Classification results are shown in Table 3. in Table 3. The proposed method achieved an accuracy of 90.32% in average. Table 4 compares the results obtained for our approach with other relevant works in the literature: Zhou et al. [4] and Ali et al. [9]. As it can be seen, the three approaches obtain similar accuracy

Table 3. Confusion matrix

	Bend	Jack	Pjump	Jump	Run	Side	Skip	Walk	Wave1	Wave2
Bend	9									
Jack		9								
Pjump			9							
Jump				7			2			
Run					7	1	1			
Side						8	1			
Skip				1	1		7			
Walk					1			8		
Wave1									9	
Wave2									1	8

Table 4. Comparative of our method against others found in the literature

	Our method	Zhou et al. [4]	Ali et al. [9]
Bend	100.0	100.0	100.0
Jack	100.0	95.7	100.0
Pjump	100.0	87.5	55.0
Jump	77.7	92.2	100.0
Run	77.7	86.7	88.8
Side	88.8	86.4	88.8
Skip	77.7	84.0	100.0
Walk	88.8	100.0	100.0
Wave1	100.0	100.0	100.0
Wave2	88.8	89.4	100.0
Reported mean accuracy	90.32	91.4	92.6

results, but our approach uses the simplest feature set and performs in real time. Performance results are not available in [4] and [9] but in our opinion they are very computational demanding methods which would be penalized for real time execution.

5 Conclusions and Future Work

This paper describes a human action recognition system based on a robust visual tracking algorithm (MAPF). Features are computed using the information available from the output of the tracking system and finally actions are classified using them. Experimental validation was performed on real videos of human actions and has shown that features extracted from a bounding box are good enough to recognize different human actions.

MAPF is a very fast algorithm as well as the proposed feature extraction method, which requires very little computation. Finally we use a SVM classifier which has been proven to be very efficient. Overall, our action recognition method is very fast because of its simplicity, allowing us to integrate it in a real time system.

As a future work, we propose to measure the achieved performance with the current features in comparison with state of the art.

Acknowledgements

This research has been partially supported by the Spanish projects TIN2008-06890-C02-02.

References

1. Wang, L., Suter, D.: Visual learning and recognition of sequential data manifolds with applications to human movement analysis. Computer Vision and Image Understanding 110, 153–172 (2008)
2. Schindler, K., Gool, L.: Action Snippets: How many frames does human action recognition require? In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008) (2008)
3. Ahmad, M., Lee, S.W.: Human action recognition using shape and CLG-motion flow from multi-view image sequences. Pattern Recognition 41, 2237–2252 (2008)
4. Zhou, H., Wang, L., Suter, D.: Human action recognition by feature-reduced Gaussian preocess classification. Pattern Recognition Letters (2009)
5. Lv, F., Nevatia, R.: Single View human Action Recognition using Key pose Matching and Viterbi Path Searching. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, June 17-22, pp. 1–8 (2007)
6. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, August 23-26, vol. 3, pp. 32–36 (2004)
7. Parameswaran, V., Chellappa, R.: View invariants for human action recognition. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 18-20, vol. 2, pp. II- 613–II-619 (2003)
8. Yan, K., Sukthankar, R., Hebert, M.: Efficient visual event detection using volumetric features. In: Tenth IEEE International Conference on Computer Vision, ICCV 2005, October 17-21, vol. 1, pp. 166–173 (2005)
9. Ali, S., Basharat, A., Shah, M.: Chaotic Invariants for Human Action Recognition. In: IEEE 11th International Conference on Computer Vision, ICCV 2007, October 14-21, pp. 1–8 (2007)
10. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, August 23-26, vol. 3, pp. 32–36 (2004)
11. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as Space-Time Shapes. Transactions on Pattern Analysis and Machine Intelligence 29(12), 2247–2253 (2007)
12. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings F Radar & Signal Processing 140 2, 107–113 (1993)
13. Moscato, P.: Memetic Algorithms: a short introduction. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 219–234. McGraw Hill, New York (1999)

14. Moscato, P., Cotta, C.: A gentle introduction to Memetic Algorithms. In: Glover, F., Kochenberger, G. (eds.) Handbook of Metaheuristics, pp. 105–144. Kluwer Academic Publishers, Boston (2003)
15. Arulampalam, S.M., Maskell, S., Gordon, N., Clapp, T.: A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Trans. on Signal Processing, 50(2), 174–178 (2002)
16. Pantrigo, J.J., Hernández, J., Sánchez, A.: Multiple and variable target visual tracking for video-surveillance applications. Pattern Recogn. Lett. 31(12), 1577–1590 (2010)
17. Hart, W.E., Krasnogor, N., Smith, J.E.: Recent Advances in Memetic Algorithms. Springer, Heidelberg (2005)
18. Carpenter, J., Clifford, P., Fearnhead, P.: Building robust simulation based filters for evolving data sets. Tech. Rep., Dept. Statist., Univ. Oxford, Oxford, U.K (1999)

Modeling and Discovering Occupancy Patterns in Sensor Networks Using Latent Dirichlet Allocation

Federico Castanedo[1], Hamid Aghajan[2], and Richard Kleihorst[3]

[1] Computer Science Department. University Carlos III of Madrid
federico.castanedo@uc3m.es
[2] Department of Electrical Engineering. Stanford University
aghajan@stanford.edu
[3] Vito & Ghent University
richard.kleihorst@vito.be

Abstract. This paper presents a novel way to perform probabilistic modeling of occupancy patterns from a sensor network. The approach is based on the Latent Dirichlet Allocation (LDA) model. The application of the LDA model is shown using a real dataset of occupancy logs from the sensor network of a modern office building. LDA is a generative and unsupervised probabilistic model for collections of discrete data. Continuous sequences of just binary sensor readings are segmented together in order to build the dataset discrete data (bag-of-words). Then, these bag-of-words are used to train the model with a fixed number of topics, also known as routines. Preliminary obtained results state that the LDA model successfully found latent topics over all rooms and therefore obtain the dominant occupancy patterns or routines on the sensor network.

Keywords: Probabilistic modeling, Sensor networks, Latent topics.

1 Introduction

The main objective of this paper is to show a way to perform probabilistic modeling of occupancy patterns from a sensor network without having any apriori or ground-truth information about the behaviors, thus following an unsupervised technique.

More specifically, this paper presents an approach based on the Latent Dirichlet Allocation (LDA) model [1] for modeling and discovering occupancy patterns in an office environment using a sensor network. LDA is a generative and unsupervised machine learning probabilistic model for collections of discrete data. In a generative machine learning algorithm, the model adjust the parameters to produce the underlying data, so the model fits to the provided data. The LDA model is one of the hierarchical Bayesian text models that has been proposed in the research community. It overcomes some of the limitations that have been reported with the probabilistic Latent Semantic Indexing (pLSI) [2], such as the over-fitting problems. Since Blei's original paper [1], LDA has been successfully

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 481–490, 2011.

applied over text corpora [3] and for classification tasks on video sequences [4]. In this work, the LDA model is applied to sequences of occupancy logs provided by a sensor network in a modern office building. The main goal is to automatically discover long-term-behavior patterns or routines on the logged data. One of the main issues of this work, compared to other approaches, is that we only employ occupancy sensors that provided binary information, indicating if the room is occupied or not. Thus, it is a technique which could be easily implemented and deployed and also respect privacy issues.

This paper covers the completely process under two different phases. First, it is necessary to obtain a good representation of the long-term activity of an specific room. Secondly, in a learning phase we answer the question of how to find the most common patterns given the training data.

In the next section some related works are described. Then, section 3 illustrates how to build the representation of the long-term activity in the sensor network. Section 4 focused on how to generate the LDA model and review it. The results of some experiments are shown in section 5. Finally, section 6 concludes the paper.

2 Related Works

With the recent advances on sensor networks it becomes more easy to obtain and process the provided data. There are a lot of research on the computer vision community which aims to model the behavior of the people being monitored. However, that research is outside of the scope of this work since we are not dealing with visual sensors. Similar approaches in the literature which covers occupancy data in an office environment could be found in the works employing the Mitsubishi Electric Research Lab (MERL) Dataset [5]. MERL dataset is a public dataset which provides around 30 million of sensor logs from a network of 200 sensors in an office environment. The dataset consist of sensors readings recorded for about 1 year at MERL building. In [6], the authors use the MERL dataset with the the aim of modeling three different social features: (1) visiting another person, (2) attending meetings with another person and (3) traveling with another person. They employed information theoretic measures (entropy) and graph cuts to obtain the previous patterns on the MERL dataset. To extract relationships on the occupancy data they model pairwise statistics over the dataset. Their goal was to identify potentially important individuals within the organization. In [7], the authors reviewed the existing techniques for the discovery of temporal patterns in sensor data and proposed a modified T-Pattern algorithm [8]. This algorithm was tested using the MERL dataset and the presented results outperformed the Lempel-Ziv compression based methods. The work of [9] also introduce temporal information in the activity recognition problem, but using temporal evidence theory. Their framework was tested using a smart home dataset.

An interesting article about consumer patterns is presented in [10]. The authors employed a wireless local area network (WLAN) to obtain the data from a shopping environment. They results show how the consumer behavior varies depending on the time of the day. In their work, they divided the consumer behavior into three segments: (1) 8:00 am to 12:00, (2) 12:00 to 16:00 and (3) 16:00 to 20:00 and presents the obtained tracking results of each segment on the shop ground plane. Our work, go further since it automatically discover the occupancy patterns on the sensor data.

Farrahi and Gatica-Perez [11] presents an interesting work in the mobile domain using also the LDA model to discover routines. The method used for generating the bag-of-words before training the model inspired the approach presented in our work. Their results also supports the advantage of using LDA for discover routines on large amount of data.

3 Building the Representation of the Long-Term Behavior Model

The long-term-behavior model is constructed using terms which are commonly employed in the information retrieval community: words, documents, topics and corpora. That is, in the context of this work *"words"* refers to occupancy patterns or location sequences over some time interval. A *"document"* refers to a complete day of words (location sequences) for a specific room. The *"corpora"* is the full data obtained from the sensors, that is the set of all documents. Finally, the *"topics"* are equivalent to the routines and the main task is to automatically infer them. So, the aim is to learn both what the topics are and which documents employ them in which proportions.

Each possible word w, of the vocabulary v, is generated taking intervals of 30 minutes which consists of three sequences of 10 minutes of occupancy over each room. The whole weekday is divided into 9 segments of day following this scheme:

1. from 00:00 to 06:00, interval 1.
2. from 06:00 to 07:00, interval 2.
3. from 07:00 to 09:00, interval 3.
4. from 09:00 to 11:00, interval 4.
5. from 11:00 to 14:00, interval 5.
6. from 14:00 to 17:00, interval 6.
7. from 17:00 to 19:00, interval 7.
8. from 19:00 to 21:00, interval 8.
9. from 21:00 to 00:00, interval 9.

So, the words generation approach of the vocabulary follow this structure: combinations of 3 sequences of **I** (someone in the room) or **O** (empty room) symbols, plus the segment time of day number (one of the previous 9), plus the room number. For instance, the word **OOO1120** refers to 30 minutes of no occupancy at the time interval 1 (from 00:00 to 6:00) for the room number 120; and the word

IOI3019 refers to an occupancy pattern of the room 19 at the time interval 3 (from 7:00 to 9:00) in which someone is in the room for 10 minutes then go out for ten minutes and come inside for at least other 10 minutes.

In the dataset employed for the experiments the vocabulary is composed of 9720 words, since there are 2^3 combinations times 9 segments times 135 rooms.

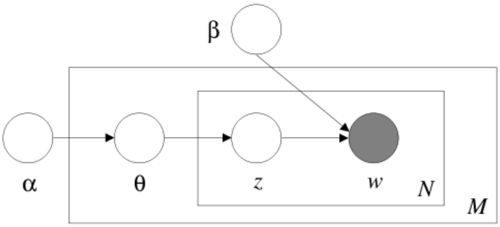


Fig. 1. Graphical representation of the LDA model, each node represents random variables, the edges possible dependence between them and the plates denote replicated structure. In reality the LDA model only observe the words (shaded w random variable) of each document.

4 Generating the LDA Model

In LDA, the word is the basic unit of discrete data and is a member of the vocabulary. More formally in LDA, a document is defined as a distribution over topics $p(document|topics)$ and the topics are defined as distributions over words $p(topics|words)$, see figure 1 for a graphical representation of the LDA model which shows the conditional Independence assumptions of the model.

LDA represents documents as random mixtures over latent topics, where each topic is characterized by a distribution over words, and assumes the following generative process for each document in a corpus:

1. Choose $N \sim Poisson(\xi)$.
2. Choose $\theta \sim Dirichlet(\alpha)$.
3. For each of the N words W_n:
   (a) Choose a topic $Z_n \sim Multinomial(\theta)$.
   (b) Choose a word W_n from $p(W_n|Z_n, \beta)$, a multinomial probability conditioned on the topic Z_n.

The dimensionality K of the Dirichlet distribution (and thus the dimensionality of the topic variable Z) is assumed known and fixed. Also, the word probabilities are parametrized by a $K \times V$ matrix β where $\beta_{ij} = p(w^j = 1|z^i = 1)$, which consists on a quantity to be estimated.

A Dirichlet distribution is a distribution over the K-dimensional probability simplex (i.e., positive vectors that sum to one):

$$\triangle_K = \{(\theta_1, ..., \theta_k) : \theta_k \geqslant 0, \sum_{i=1}^{k} \theta_i = 1\} \quad (1)$$

$(\theta_1, ..., \theta_k)$ is known to be Dirichlet distributed:

$$(\theta_1, ..., \theta_k) \sim Dirichlet(\alpha_1, ..., \alpha_k) \tag{2}$$

with parameters $(\alpha_1, ..., \alpha_k)$, if has the following probability density:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} ... {\theta_k}^{\alpha_{k-1}} \tag{3}$$

where the parameter α is a k-vector with components $\alpha_i > 0$ and $\Gamma(x)$ is the Gamma function.

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of K topics z, and a set of N words w is given by:

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(Z_n|\theta) p(w_n|Z_n, \beta) \tag{4}$$

In order to obtain the parameters of equation (4), it is necessary to employ estimation sampling techniques since it is intractable to compute in general due to the coupling between θ and β. Different approximate inference algorithms could be employed, such as Gibbs sampling, Laplace approximation, variational approximation and Markov Chain Monte Carlo [12]. For a good comparison between the different inference algorithms, see [13]. We followed the approach detailed in [1] and used a variational *Expectation-Maximization* inference algorithm in order to find the parameters α, θ and β.

These parameters provide information about the underlying data (corpus). The parameter α indicates how semantically diverse documents are (in our case daily occupancy patterns on each room), with lower alpha values indicating increase diversity. The parameter $\beta_{1:k}$ provides information about how similar the different topics are, since it provides the log-likelihood of $prob(word|topic)$ for each one of the K topics.

When a LDA model is trained it is possible to infer the multinomial distributions θ which provides the probability of each topic in a new document. These distributions can then be used for classifying new documents or measuring similarities between documents.

5 Experimental Results

In this section, we present preliminary results of the previous model. The data for our experiment was obtained from monitoring the Innotek building in Geel, Belgium. This 3 floor building facilitates start-up companies. It contains offices with different uses. In the building we will find Innotek's own administrative offices, such as receptionist, secretarial office, management office, building maintenance office, cantine and restroom facilities. Also, several rooms are rented out to start-up companies, usually 1 to 4 rooms per company. These rooms are mainly functioning as office space to meet prospective customers or guests, so they show a different character of use compared to regularly occupied offices.

The top-floor of the Innotek building houses some laboratories where people have to be present to follow up experiments. These spaces also show occupancy at night, during weekends and in holidays.

The occupancy data is captured by centrally mounted high-quality PIR-based sensors that are part of the Philips system to install the automatic lighting. Detection of motion will trigger the lights which will remain switched on for at least 10 minutes. If no motion has been detected in this period, the lights will turn-off again. Innotek has arranged for connecting these sensors to the bus of their Johnson Controls heating management system, which enables their use to control the room temperature also. The climate management system will set the room to ”hibernation mode” if no occupancy has been detected for a certain time interval. Once occupancy is detected, the room climate will go to the comfort level as set locally per room by the inhabitants. New office buildings in Flanders will have to meet smart building controls like this in the very near future.

As the Johnson Controls sensor bus can be read out at a central point we have been able to log the data of most rooms of the building on a 1 minute accurate timescale. The events are based on integration of detections over the 1 minute interval and indicate the occupancy with a high level of confidence.

The plans of the three floors of the Innotek building are shown in figure 2. We worked on the logs of the Innotek dataset related to the period from March 12 2010 to August 28 2010. The log file follows this structure:

1. 0 01 0 10 3 12 6 13 37
2. 0 01 1 10 3 12 6 14 33
3. 0 01 0 10 3 12 6 14 48
4. 0 01 1 10 3 12 6 15 5
5. ...

The first line means that sensor 1 of floor 0 provides no occupancy information (0 at third column) the day 12 of march 2010, which has a type day of 6 (meaning it was Friday) at the time 13:37. In the second line, the same sensor provides occupancy information (value 1 at third column) at time 14:33, line three shows that the room was empty again at time 14:48 and so on.

With this information (only binary occupancy with 1 minute frequency) we construct the documents, one document for each room and day, which are sequences of words over 30 minutes interval. The frequency patterns can be seen more clear in a video, which plots the obtained frequencies on the building ground-plane, posted online at `http://fcastanedo.com/wp-content/uploads/2010/11/foo-candidate-30min.avi`.

A document is formed using the frequency of words which composed them, following an approach similar to *tf-idf* [14] scheme. Therefore the original data is transformed on another which follows this structure:

1. 9 0:34 8:4 16:10 24:10 32:16 40:16 48:10 56:10 64:16
2. 23 0:34 8:4 16:2 17:2 18:1 19:1 20:1 23:3 31:10 39:16 40:7 41:1 43:1 44:1 46:1 47:5 50:1 52:1 53:1 54:1 55:6 56:10 64:16
3. ...

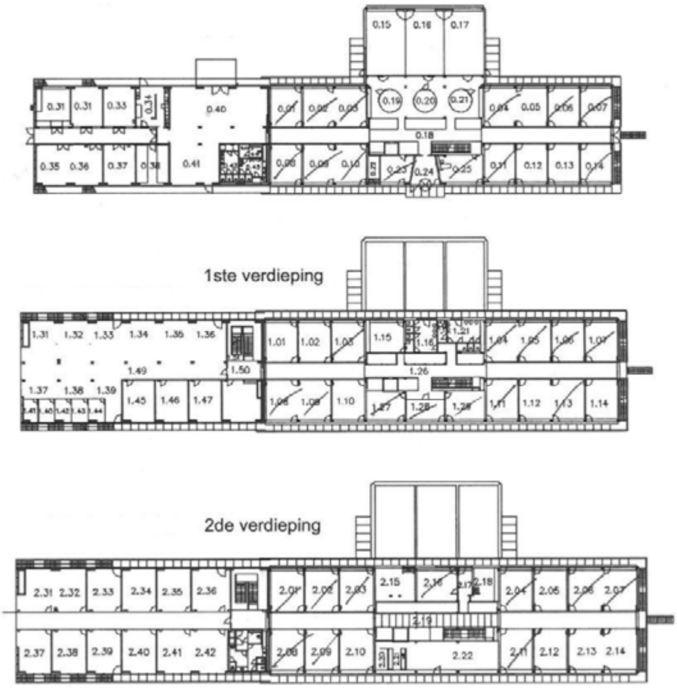


Fig. 2. Plan of the three floor Innotek building

Each line is called a *document*, where the first value (M) gives the number of different words in the document, then there are M pairs of I:J numbers which I indicates the index of the word in the vocabulary V and J the count frequency of this word in the document. Remember, that in our context a document means an occupancy pattern of a room over one day. In these experiments we skip the weekend days (Saturday and Sunday) since, as you can see in figure 3, do not provide meaningful information. Also, we are more interested in the weekdays patterns due to the underlying nature of the data. The words generated from segment (***1***) and from segment 9 (***9***) are not taken into account for these experiments since there are many of them and we think that do not provide useful information. These kind of words are known in the natural language processing domain as stop-words.

More specifically the dataset used for training the model is composed of 22 weeks and 5300 documents with 7840 different terms and 66577 total words in all the documents.

In table 1 some of the obtained 90 topics with the 5 most probable words of each topic are shown. Topics 7, 10, 15, 24, 52, 53, 74 and 76 provide a similar occupancy pattern of rooms 70, 103, 69, 8, 9, 96, 7 and 11 respectively. For instance in rooms 7,8 and 9, which corresponds to rooms 0.08, 0.09 and 0.10 of the first floor, the words III6 and OOO8 are the most probable ones (with

Table 1. Some obtained topics and the 5 most probable words for each topic (with $k = 90$)

Topic 7		Topic 10		Topic 15		Topic 24		Topic 52	
words	p(w|topic)	words	p(w|topic)	words	p(w|topic)	words	p(w|topic)	words	p(w|topic)
III6070	0.211401	III6103	0.178530	III6069	0.199488	III6008	0.219808	III6009	0.210557
OOO8070	0.131730	OOO8103	0.132126	OOO8069	0.128993	OOO8008	0.118008	OOO8009	0.123193
IIO7070	0.022237	IOO7103	0.021666	OII3069	0.014108	IIO7008	0.013609	IOO7009	0.011962
OII3070	0.013220	IIO7103	0.021666	IOO7069	0.012204	III8008	0.005343	IIO7009	0.011367
IOO7070	0.013220	III7103	0.013787	IIO7069	0.011843	IOI7008	0.003309	IIO3009	0.004305
Topic 53		Topic 74		Topic 76		Topic 36		Topic 11	
words	p(w|topic)	words	p(w|topic)	words	p(w|topic)	words	p(w|topic)	words	p(w|topic)
III6096	0.204948	III6007	0.213525	III6011	0.209088	OOO8007	0.231541	III4023	0.176400
OOO8096	0.131598	OOO8007	0.101672	OOO8011	0.107636	OOO2054	0.193013	OOO7023	0.154432
IIO7096	0.016975	IIO7007	0.013434	III7011	0.042809	III5069	0.017457	OOO8023	0.148823
OII3096	0.015717	III8007	0.007506	OOI8011	0.012842	OOO5102	0.016907	IOO6023	0.030838
IOO7096	0.015717	IIO8007	0.006711	III8011	0.004283	III6069	0.012438	IOO7023	0.027737

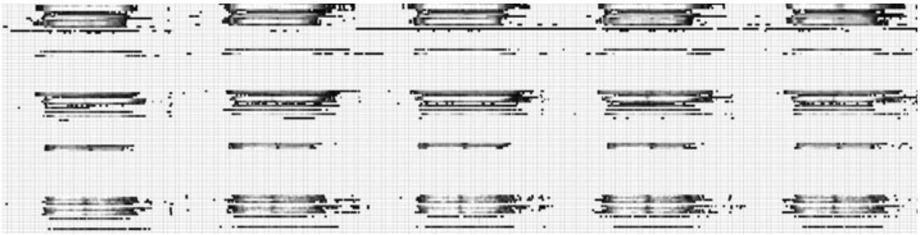


Fig. 3. Big picture of occupancy frequencies for all the rooms over the whole data. Each cell corresponds to a 10 minute interval. Rows indicate specific rooms and columns the time evolution.

a prob>(0.3)) meaning that this rooms are occupied between 14:00 and 17:00 and empty between 19:00 and 21:00. The topic 36 indicates that when room 7 is empty from 19:00 to 21:00, room 54 is also empty from 6:00 to 7:00. Finally, the topic 11 shows a correlation when the room 23 is occupied from 9:00 to 11:00 and empty from 17:00 to 21:00 (OOO7023 and OOO8023).

6 Conclusions and Future Work

The proposed method for modeling the occupancy behaviour of an office building is based on the "bag of words" assumption, so each word in a document follows the ex-changeability assumption. The ex-changeability assumption means that the words are conditionally independent and identically distributed with respect to a latent parameter (topics).

A probabilistic model is enough powerful to handle the uncertainty of the underlying data. The main advantage of the LDA model over other techniques, is that it allows to obtain a probability distribution of several words in each topic, following an unsupervised scheme. The obtained results show that the LDA model is enough powerful to handle the detection of complex patterns over a long-term dataset. In the future we would like to obtain most relevant results for the building occupancy environment.

With the purpose of stimulate similar works using this dataset, we plan to provide the data and the developed vocabulary, to researchers interested in running further analysis.

Acknowledgments

The first author would like to thank the 2010 UC3M grant: *Ayudas de movilidad del programa propio de investigación* and projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC and CAM CONTEXTS 2009/ TIC-1485 for partially support this research.

The authors want to thank Erik Degroof and Luc Peeters of Innotek Belgium for their cooperation and the use of their dataset.

References

1. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
2. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57. ACM, New York (1999)
3. Griffiths, T., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America 101(suppl. 1), 5228–5235 (2004)
4. Niebles, J., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. International Journal of Computer Vision 79(3), 299–318 (2008)
5. Wren, C.R., Ivanov, Y.A., Leigh, D., Westhues, J.: The merl motion detector dataset. In: Proceedings of the 2007 workshop on Massive datasets, Ser. MD 2007, pp. 10–14. ACM, New York (2007), http://doi.acm.org/10.1145/1352922.1352926
6. Connolly, C., Burns, J., Bui, H.: Recovering social networks from massive track datasets. In: IEEE Workshop on Applications of Computer Vision, WACV 2008, pp. 1–8. IEEE, Los Alamitos (2008)
7. Salah, A., Pauwels, E., Tavenard, R., Gevers, T.: T-Patterns Revisited: Mining for Temporal Patterns in Sensor Data. Sensors 10(8), 7496–7513 (2010)
8. Magnusson, M.: Discovering hidden time patterns in behavior: T-patterns and their detection. Behavior Research Methods Instruments and Computers 32(1), 93–110 (2000)
9. McKeever, S., Ye, J., Coyle, L., Bleakley, C., Dobson, S.: Activity recognition using temporal evidence theory. Journal of Ambient Intelligence and Smart Environments 2(3), 253–269 (2010)
10. Skogster, P., Uotila, V., Ojala, L.: From mornings to evenings: is there variation in shopping behaviour between different hours of the day? International Journal of Consumer Studies 32(1), 65–74 (2008)

11. Farrahi, K., Gatica-Perez, D.: Discovering routines from large-scale human locations using probabilistic topic models. ACM Transactions on Intelligent Systems and Technology 2(1) (2011)
12. Jordan, M., Ghahramani, Z., Jaakkola, T., Saul, L.: An introduction to variational methods for graphical models. Machine learning 37(2), 183–233 (1999)
13. Asuncion, A., Welling, M., Smyth, P., Teh, Y.W.: On smoothing and inference for topic models. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (2009)
14. Salton, G., McGill, M.: Introduction to modern information retrieval, New York (1983)

Multicamera Action Recognition with Canonical Correlation Analysis and Discriminative Sequence Classification

Rodrigo Cilla, Miguel A. Patricio, Antonio Berlanga, and José M. Molina

Computer Science Department. Universidad Carlos III de Madrid
Avda. de la Universidad Carlos III, 22. 28270 Colmenarejo, Madrid, Spain
{rcilla,mpatrici}@inf.uc3m.es, {berlanga,molina}@ia.uc3m.es

Abstract. This paper presents a feature fusion approach to the recognition of human actions from multiple cameras that avoids the computation of the 3D visual hull. Action descriptors are extracted for each one of the camera views available and projected into a common subspace that maximizes the correlation between each one of the components of the projections. That common subspace is learned using Probabilistic Canonical Correlation Analysis. The action classification is made in that subspace using a discriminative classifier. Results of the proposed method are shown for the classification of the IXMAS dataset.

1 Introduction

The recognition of human actions has received an increasing attention by the computer vision community during the last years [10]. One of the current trends in the field is how to efficiently combine the perceptions grabbed from different viewpoints in order to create more robust action recognition systems. This way the system can cover wider scenes, being able to deal with the possible occlusions caused by walls and furniture that would make the recognition from a single view very difficult if not impossible.

Although there has been different proposals of human action recognition systems at the different sensor fusion levels proposed by Dasarathy [5] as [4,15] in the decision-in decision-out level or [22,12] at the feature-in decision-out level, the most successful approaches have been defined at the feature-in feature-out level. These approaches extract human silhouettes from the different cameras using for example background subtraction [16], and then reconstruct the 3D visual hull of the human [9] as the feature to be used for the recognition. This way, Weinland et al. [21] have proposed the Motion History Volumes (MHV) as an extension of the popular Motion History Image (MHI) [3] to 3D. Action classification is then made using Fourier analysis of the MHV. Peng et al [13] have performed multilinear analysis of the voxels in the visual hull. Turaga et al. [19] have studied the visual hulls using Stiefel and Grassman manifolds, reporting the best results for action recognition in 3D until date. The main drawback of these methods is that 3D visual hull reconstruction has a high computational

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 491–500, 2011.

burden and requires accurate calibration parameters of each one of the cameras observing the scene. Also, the computation of the 3D visual hull requires at least the silhouettes from 2 different camera viewpoints.

This work presents a novel method for the recognition of human actions using multiple cameras at the feature fusion level but without explicitly reconstructing the visual hull or other 3D descriptor. Experimental results reported in the literature [19,13] have shown that visual hulls can be projected into low dimensional manifolds where most of their variance is preserved. Moreover, a silhouette is the projection of a visual hull into the camera plane, and different works [20] have also reported that they can be parametrized into low dimensional manifolds. The aim of our method is to find a set of projection functions, one for each camera, that project the corresponding silhouettes into a common low dimensional manifold. We think that the representation of the silhouettes into that common low dimensional manifold would be equivalent to the low dimensional representation fo the visual hull, so similar results can be achieved in human action recognition.

Probabilistic continuous latent variable models provide a framework for manifold learning where low and high dimensional representations are related via the factorization of their joint probability distribution. We test the usage of the Probabilistic Canonical Correlation Analysis (PCCA) model [2] to learn the projections of the features observed at the different cameras into a subspace that maximizes the correlation between their components. The representation of the observed features into that subspace is then used for action sequence classification.

Paper is organized as follows: section 2 presents how the proposed system is structured; 3 reviews the Canonical Correlation Analysis model; section 4 describes the sequence classifier that is going to be used to test the system; section 5 shows some experimental validation of the method; finally, section 4 discusses the conclusions and future lines of the work.

2 System Overview

The architecture of the proposed system is shown on figure 1. The images grabbed by the C different cameras observing the scene are independently processed to extract a sequence of action descriptors $X_c = x_1c, \ldots, x_{Tc}$, $1 \leq c \leq C$, and T is the total number of frames grabbed. The C sequences of actions descriptors extracted are fused projecting them into a common subspace to give a sequence of common action descriptors $Z = z_1, \ldots, z_T$, $z_t = F\left(x_{t1}, \ldots, x_{tC}\right)$. Finally each sequence is introduced into an action classifier to make the decision on the action being performed in the sequence.

3 Canonical Correlation Analysis

Canonical Correlation Analysis is the method we use for the fusion of action descriptors. In the next paragraphs we give an overview of the classical and the probabilistic formulation.

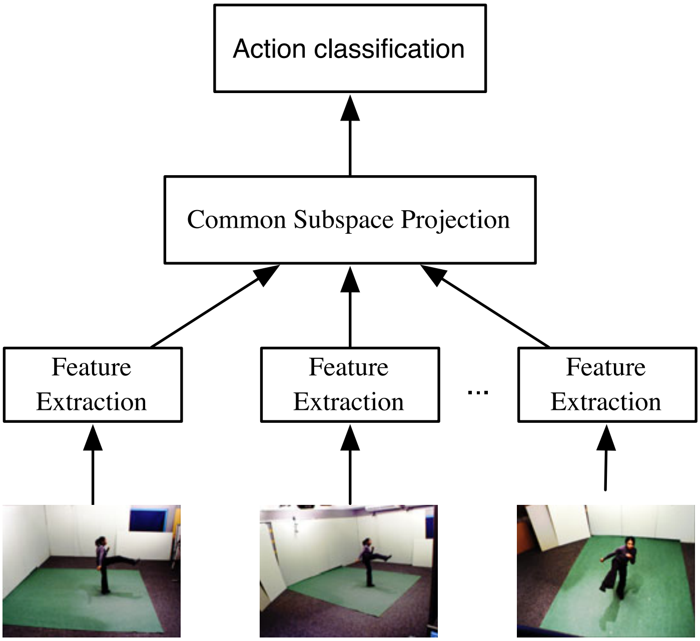


Fig. 1. Overview of the proposed system. Features are extracted for each available view. They are projected into a common subspace by Canonical Correlation Analysis. This projection is the used for action classification.

3.1 Classical Definition

Canonical Correlation Analysis [6] allows measuring the linear relationship between a pair of multidimensional variables. Given two random variables x_1 and x_2 of dimension d^1 and d^2 and zero mean, CCA finds a pair of linear transformations w_1, w_2, such that one component within each set of transformed variables is correlated with a single component in the other set. The correlation between the corresponding components is called canonical correlation, and there can be at most $d = min\,(d_1, d_2)$ canonical correlations. The first canonical correlation is defined as:

$$\begin{aligned}\rho &= \max_{w1,w2} \frac{\left\langle w_1^T x_1 \cdot w_2^T x_2 \right\rangle}{\sqrt{\left\langle \|w_1^T x_1\|^2 \right\rangle \left\langle \|w_2^T x_2\|^2 \right\rangle}} \\ &= \max_{w1,w2} \frac{w_1^T \left\langle x_1 x_2^T \right\rangle w_2}{\sqrt{w_1^T \left\langle x_1 x_1^T \right\rangle w_1 w_2^T \left\langle x_2 x_2^T \right\rangle w_2}}\end{aligned}$$

where $\left\langle x_1 x_1^T \right\rangle$, $\left\langle x_2 x_2^T \right\rangle$ and $\left\langle x_1 x_2^T \right\rangle$ are respectively estimated as $\tilde{\Sigma}_{11}$, $\tilde{\Sigma}_{22}$ and $\tilde{\Sigma}_{12}$, i.e, the different minors of the sample covariance matrix $\tilde{\Sigma} = \begin{pmatrix} \tilde{\Sigma}_{11} & \tilde{\Sigma}_{12} \\ \tilde{\Sigma}_{21} & \tilde{\Sigma}_{22} \end{pmatrix}$

of a set of training data $x = (x_1, x_2)$. The rest of canonical correlation directions are orthogonal to w_1 and w_2 respectively. They can be computed as the solutions of the generalized eigenvalue problem:

$$\begin{pmatrix} \tilde{\Sigma}_{11} & \tilde{\Sigma}_{12} \\ \tilde{\Sigma}_{21} & \tilde{\Sigma}_{22} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = (1+\rho) \begin{pmatrix} \tilde{\Sigma}_{12} & 0 \\ 0 & \tilde{\Sigma}_{21} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

The classical CCA model is defined for only two random variables x_1 and x_2. Bach and Jordan [1] generalize it to m random variables. The generalized eigenvalue problem to solve is then defined as:

$$\begin{pmatrix} \tilde{\Sigma}_{11} & \cdots & \tilde{\Sigma}_{1m} \\ \vdots & & \vdots \\ \tilde{\Sigma}_{m1} & \cdots & \tilde{\Sigma}_{mm} \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix} = \lambda \begin{pmatrix} \tilde{\Sigma}_{11} & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \tilde{\Sigma}_{mm} \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix}$$

3.2 Probabilistic Interpretation

Bach and Jordan [2] made a probabilistic interpretation of CCA extending the probabilistic interpretation of PCA proposed by Tipping and Bishop [17]. They define the following generative model, also shown on figure 2:

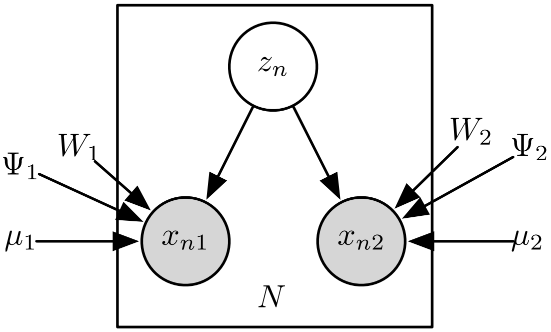


Fig. 2. Graphical model of the probabilistic interpretation of CCA made by Bach and Jordan [2] for two variables

$$\begin{aligned} z_n &\sim \mathcal{N}(0, I_q) \\ x_{n1} &\sim \mathcal{N}(W_1 z_n + \mu_1, \Psi_1) \\ x_{n2} &\sim \mathcal{N}(W_2 z_n + \mu_2, \Psi_2) \end{aligned}$$

They also show that the maximum likelihood estimates of the model parameters are given by:

$$\hat{W}_1 = \tilde{\Sigma}_{11} U_1 M_1 \tag{1}$$

$$\hat{W}_2 = \tilde{\Sigma}_{22} U_2 M_2 \tag{2}$$

$$\hat{\Psi}_1 = \Sigma_{11} - \hat{W}_1 \hat{W}_1^T \tag{3}$$

$$\hat{\Psi}_2 = \Sigma_{22} - \hat{W}_2 \hat{W}_2^T \tag{4}$$

$$\hat{\mu}_1 = \tilde{\mu}_1 \tag{5}$$

$$\hat{\mu}_2 = \tilde{\mu}_2 \tag{6}$$

where M_1 and M_2 are arbitrary matrices such that $M_1 M_2^T = P_q$, being P_q a matrix with the canonical correlations on its diagonal. U_{1q} and U_{2q} are the first q canonical directions.

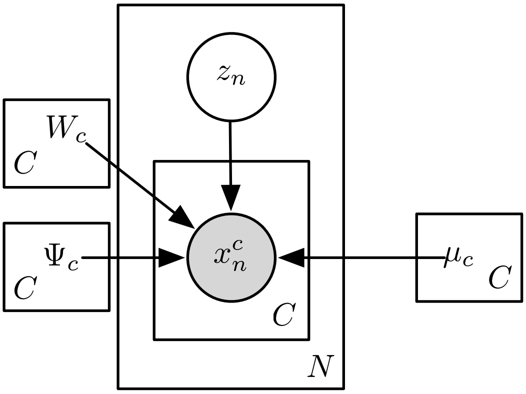


Fig. 3. Generalization of PCCA to C different data sources used in this paper

This model is easily generalizable to C different sources of data. The graphical model in that case corresponds to the shown on figure 3. Given a set of C different sources, each source c is generated as:

$$\begin{aligned} z_n &\sim \mathcal{N}(0, I_q) \\ x_{nc} &\sim \mathcal{N}(W_c z_n + \mu_c, \Psi_c) \end{aligned}$$

The maximum likelihood estimates of the parameters are then given by:

$$\hat{W}_c = \tilde{\Sigma}_{cc} U_{cd} M_c \tag{7}$$

$$\hat{\Psi}_c = \Sigma_{cc} - \hat{W}_c \hat{W}_c^T \tag{8}$$

$$\hat{\mu}_c = \tilde{\mu}_c \tag{9}$$

This probabilistic generalization of the CCA model is employed in our system to combine the feature descriptors extracted from the different views. We choose the probabilistic interpretation as it would allow us to easily integrate the model as a part of larger graphical models for action recognition.

4 Hidden Conditional Random Fields

Hidden Conditional Random Fields (HCRF) [14] extend Conditional Random Fields [8] introducing hidden state variables into the model. A HCRF is an undirected graphical model composed of three different set of nodes, as figure 4 shows. The node y represents the labelling of the input sequence. $X = x_1, \ldots, x_t$ is the set of nodes corresponding to the sequence observations $H = h_1, \ldots, h_t$ is the set of hidden variables modelling the relationship between the observations x_i and the class label y and the temporal evolution of the sequence

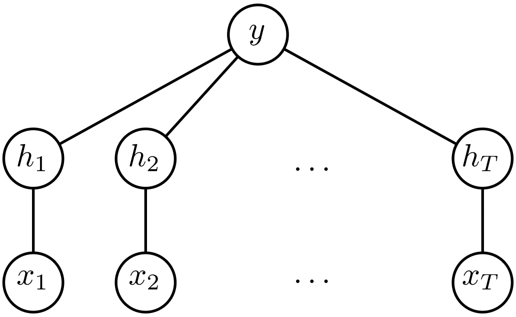


Fig. 4. Graphical model representation of the Hidden Conditional Random Field

The conditional probability of a sequence label y and a set of hidden part assignments $\mathbf{h}$ given a sequence of observations X is defined using the Hammersley-Clifford theorem of Markov Random Fields:

$$P\left(y, \mathbf{h} \mid \mathbf{x}, \theta\right) = \frac{e^{\Psi(y,\mathbf{h},\mathbf{x};\theta)}}{\sum_{y'} \sum_{h} e^{\Psi(y,\mathbf{h},\mathbf{x};\theta)}} \tag{10}$$

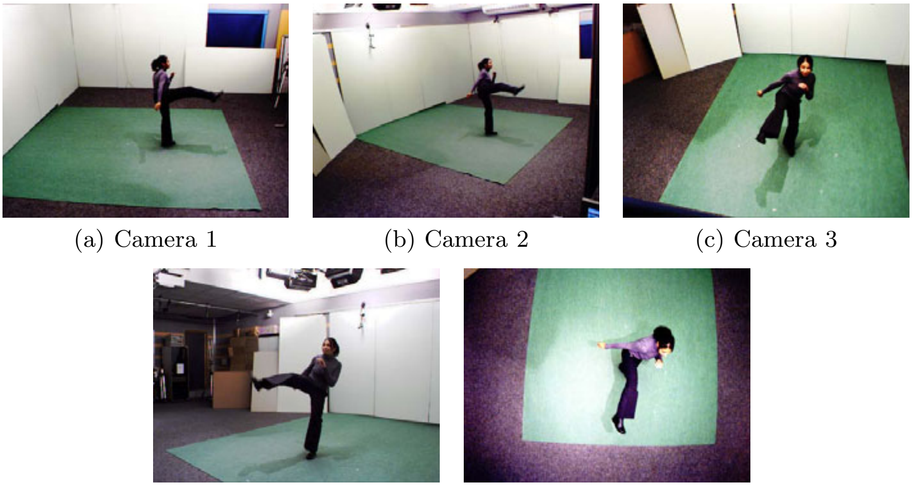
where θ is the vector of model parameters. The conditional probability of the class label y given the observation sequence X is obtained marginalizing over all the possible assignments of hidden parts $\mathbf{h}$:

$$P(y \mid \mathbf{x}, \theta) = \frac{\sum_{h} e^{\Psi(y,\mathbf{h},\mathbf{x};\theta)}}{\sum_{y'} \sum_{h} e^{\Psi(y\prime,\mathbf{h},\mathbf{x};\theta)}} \tag{11}$$

The potential function $\Psi\left(y, \mathbf{h}, \mathbf{x}; \theta\right)$ is a linear function of the input:

$$\begin{aligned} \Psi\left(y, \mathbf{h}, \mathbf{x}; \theta\right) = & \sum_{i} \phi\left(x_i\right) \cdot \theta(h_i) + \sum_{i} \theta\left(y, h_i\right) \\ & + \sum_{(j,k)\in E} \theta\left(y, h_j, h_k\right) \end{aligned} \tag{12}$$

The first term, parametrized by $\theta\left(h_i\right)$ measures the compatibility of each observation x_i with the hidden variable h_i. The second term measures the compatibility of the hidden part h_i with the class label and is parametrized by $\theta((y, h_i)$. Finally, the third term models sequence dynamics, measuring the compatibility of adjacent hidden parts h_i and h_j with the class y.



(a) Camera 1 (b) Camera 2 (c) Camera 3

(d) Camera 4 (e) Camera 5

Fig. 5. The *kick* action in the IXMAS dataset from the five available views

Given a set of training samples $\{x^i, y^i\}$, model parameters are adjusted maximizing the L_2 regularized conditional likelihood function of the model:

$$L(\theta) = \sum_{i=1}^{n} \log P(y_i \mid \mathbf{x}_i, \theta) + \frac{||\theta||^2}{2\sigma} \tag{13}$$

The optimal parameters θ^* maximizing the conditional likelihood function are found using Quasi-Newton gradient based methods. Both the computation of the posterior probability on equation 11 and the auxiliary distributions that appear on the gradient of 13 can be efficiently made using belief propagation,as proposed in [14].

5 Experiments

5.1 Experimental Setup

The proposed algorithms are going to be tested in the classification of IXMAS dataset [21]. This dataset contains 11 actions performed by 10 different actors at least 3 times each. The actions are recorded from 5 different viewpoints. The algorithms are going to be tested using Leave-One-Actor-Out Cross Validation (LOAO-CV): The algorithms are trained with all the actors unless one, used for validation.

The system is going to be tested using the action descriptor proposed by Tran et al. [18], combining optical flow and appearance information. It is used in the system because it has shown a high experimental performance. The bounding box of a human being is normalized to a square box, from which human shape and optical flow are computed. Vertical and horizontal planes of the optical flow are split and blurred. A radial histogram is computed over each of the optical flow planes and the shape. The three histograms are concatenated into 216-d vector. Lastly, PCA reduction of the surrounding past and future vectors is appended to finally generate a descriptor of $D_{TRAN} = 286$ dimensions. Readers are referred to [18] for more details.

In order to sped-up the CCA computation, PCA analysis of the descriptors is performed, retaining only the 100 principal components. CCA is going to be trained for latent dimensionality values of $q = 10, 12, 14, 16, 18, 205$. The number of hidden states of the HCRF is going to be fixed to $|H| = 11, 22$.

5.2 Results and Discussion

Table 1 shows the results obtained by the proposed method. It can be seen that the maximum accuracy is obtained for $q = 20$ dimensions and 22 hidden states. There accuracy starts growing with the number of dimensions, to then decrease and again increase to achieve teh best results. These phenomena gives an idea of how difficult is to manually parametrize the proposed methods

Table 1. Results obtained for different sizes of latent dimensionality

# dims $\lvert H\rvert$	10	12	14	16	18	20
11	80.78	81.69	81.38	81.08	78.68	81.38
22	80.78	81.19	83.78	83.18	82.58	85.59

Finally, table 2 compares the results of our method to others. While it improves the results achieved by other methods, it is still far from the results obtained by the methods based on 3D visual hulls.

Table 2. Comparison of the accuracy of our method to others

Method	Accuracy	Type
Srivastava et al. [15]	81.4	Decision-in Decision-out
Our	85.59	Feature-in Feature-out
Weinland et al. [21]	93.33	2D Feature-in 3D Feature-out
Peng et al. [13]	94.59	2D Feature-in 3D Feature-out

6 Conclusions

This work has shown a preliminary approach to the fusion of features for human action recognition using a subspace learning technique. Feature descriptors extracted from different camera views have been projected into a common subspace learned using Canonical Correlation Analysis. The action classification has been made in that subspace. Although the results achieved have been inferior to the obtained by state of the art 3D methods, we belief a non linear extension of the method using a mixture of Canonical Correlation Analyzers would reduce the gap [7].

Other strategy to test in the future would be to integrate the PCCA model into a sequence manifold learning method such the introduced in [11], in order to use the temporal evolution of the features for subspace regularization. Finally, other strategy to try would be to integrate the action classification with the resulting model, to perform the learning of the dimensionality reduction and the action classes at the same time.

References

1. Bach, F., Jordan, M.: Kernel independent component analysis. The Journal of Machine Learning Research 3, 1–48 (2003)
2. Bach, F., Jordan, M.: A probabilistic interpretation of canonical correlation analysis. Dept. Statist., Univ. California, Berkeley, CA, Tech. Rep 688 (2005)
3. Bobick, A., Davis, J.: The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(3), 257–267 (2001)
4. Cilla, R., Patricio, M.A., Berlanga, A., Molina, J.M.: Fusion of single view soft k-NN classifiers for multicamera human action recognition. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS, vol. 6077, pp. 436–443. Springer, Heidelberg (2010)
5. Dasarathy, B.: Sensor fusion potential exploitation-innovative architectures and illustrative applications. Proceedings of the IEEE 85(1), 24–38 (2002)
6. Hardoon, D., Szedmak, S., Shawe-Taylor, J.: Canonical correlation analysis: an overview with application to learning methods. Neural Computation 16(12), 2639–2664 (2004)
7. Klami, A., Kaski, S.: Local dependent components. In: Proceedings of the 24th international conference on Machine learning, pp. 425–432. ACM, New York (2007)
8. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: International Conference on Machine Learning (2001)
9. Laurentini, A.: The visual hull concept for silhouette-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence, 150–162 (1994)
10. Lavee, G., Rivlin, E., Rudzsky, M.: Understanding Video Events: A Survey of Methods for Automatic Interpretation of Semantic Occurrences in Video. IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews 39(5), 489–504 (2009)
11. Li, R., Tian, T., Sclaroff, S.: Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series. In: IEEE 11th International Conference on Computer Vision, ICCV 2007, pp. 1–8. IEEE, Los Alamitos (2007)
12. Määttä, T., Härmä, A., Aghajan, H.: On efficient use of multi-view data for activity recognition. In: 4th IEEE/ACM International Conference on Distributed Smart Cameras, ICSDC 2010, pp. 158–165 (2010)
13. Peng, B., Qian, G., Rajko, S.: View-Invariant Full-Body Gesture Recognition via Multilinear Analysis of Voxel Data. In: Third ACM/IEEE Conference on Distributed Smart Cameras (September 2009)

14. Quattoni, A., Wang, S., Morency, L.P., Collins, M., Darrell, T.: Hidden conditional random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(10), 1848–1852 (2007)
15. Srivastava, C., Iwaki, H., Park, J., Kak, A.C.: Distributed and Lightweight Multi-Camera Human Activity Classification. In: Third ACM/IEEE Conference on Distributed Smart Cameras, pp. 1–8 (September 2009)
16. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, Los Alamitos (2002)
17. Tipping, M., Bishop, C.: Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 61(3), 611–622 (1999)
18. Tran, D., Sorokin, A.: Human activity recognition with metric learning. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 548–561. Springer, Heidelberg (2008)
19. Turaga, P., Veeraraghavan, A., Chellappa, R.: Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE, Los Alamitos (2008)
20. Wang, L., Suter, D.: Visual learning and recognition of sequential data manifolds with applications to human movement analysis. Computer Vision and Image Understanding 110(2), 153–172 (2008)
21. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. Computer Vision and Image Understanding 104(2-3), 249–257 (2006)
22. Wu, C., Khalili, A., Aghajan, H.: Multiview Activity Recognition in Smart Homes with Spatio-Temporal Features. In: 4th IEEE/ACM International Conference on Distributed Smart Cameras, ICSDC 2010, pp. 142–149 (2010)

Low-Power Bed / Seat Occupancy Sensor Based on EMFi

Francisco Fernandez-Luque[1,2], Juan Zapata[1], and Ramón Ruiz[1]

[1] Depto. Electrónica, Tecnología de Computadoras y Proyectos
ETSIT- Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad Politécnica de Cartagena
Antiguo Cuartel de Antigones. Plaza del Hospital 1, 30202 Cartagena, Spain
{juan.zapata,ramon.ruiz}@upct.es
http://www.detcp.upct.es

[2] Ambient Intelligence & Interaction SLL. Edif. CEEIM mod. 11, 30100 Espinardo, Murcia, Spain
ff.luque@ami2.net

Abstract. Ambient Assisted Living (AAL) systems demand is raising. The use of bed / seat occupancy sensors is imperative for this kind of ubiquitous monitoring systems. Pressure mats are a first way to solve this feature, but several environmental dependencies make them weak to be an efficient and reliable solution for large volume deployments. Solutions based on force-to-resistor transducer seems to imply a too high power consumption to be integrated on wireless sensor nodes. A force-capacitive transducer based sensor has been proposed, implemented and tested in this paper. This sensor, based on Electro-Mechanical Films (EMFi) is able to detect force variations in a quasi-passive way. This detection is used to trigger an active mechanism to measure the weight by means of the transducer capacity. A low-power wireless sensor node prototype including this new sensor has been assembled and tested with a wide range of weights. The occupancy detection was successful and the power consumption of the node was increased at less that a 15%.

Keywords: seat, chair, bed, occupancy, EMFi, AAL, ubiquitous monitoring, WSN.

1 Introduction

Increasing health care costs and an aging population are placing significant strains upon the health care system. Small pilot studies have shown that meeting seniors' needs for independence and autonomy, coupled with expanded use of home health technologies, and provide improved assistential outcomes. Difficulty with reimbursement policies, governmental approval processes, and absence of efficient deployment strategies has hampered adopting non-obtrusive intelligent monitoring technologies.

In this field, DIA project - Intelligent Warning Device, from Spanish *Dispositivo Inteligente de Alerta* - aims to develop devices which detect behaviour patterns from their users and use them to take alert actions when significant variations

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 501–510, 2011.

happened. This project is leaded by the company Ambiental Intelligence and Interaction S.L.L. (Ami2) [1] in collaboration with the Technical University of Cartagena - UPCT, from *Spanish Universidad Politécnica de Cartagena* -, at the sensory layer of the system [2], and with the University of Murcia - UMU, from Spanish *Universidad de Murcia* - at the reasoning and context extraction layers [3].

The occupancy detection in bed or seat is useful in ambient assited living (AAL) systems [4]. The developed DIA system incorporates an implementation of these detectors based on the contact mats deployed under the mattress, in the case of bed, or under a cushion, in the case of the chair or couch [2]. This system has presented problems difficult to solve without a change in technology, since their functionality is dependent on various environmental factors such as the type of mattress, mattress position or weight of the user. Other recent work in the same field have opted for the use of sensors based on resistive transducers [5], [6]. An example of this type of sensor is based on the FlexiForce transducer from the manufacturer Tekscan [7]. These devices may be suitable for support AAL systems, except for their high energy consumption.

In this paper, we propose to solve this problem by using MEMS (*Micro-Electro-Mechanical Systems*). In particular, EMFi based transducers (*Electro-Mechanical Film* [8]) can be used to implement foil capacitors, whose capacity is dependent on the pressure supported by the surface.

The application scenario and, specifically, the occupancy detection issue are approached in Section 2. The design and implementation, on a prototype, of the new sensor is described in Section 3. Results obtained from the different modules and the entire device are shown in Section 4. Finally, some conclusions and possible future work lines are refered in Section 5.

2 Approaching the Occupancy Detection Issue

2.1 Application Scenario

A first prototype scenario has been developed in which a user will have a home assistance system that is able to monitor his or her activity in order to detect incidents and uncommon activities. The prototype house or scenario has a bedroom, a hall, a corridor, a toilet, a kitchen, and a living room.

DIA, the current version of the developed intelligent warning device is a minicomputer-based system connected to a network of sensors deployed on walls, doors and furniture in the house of the user. The device detects excessive idle times, or an anomalous location that may mean that the person has been dropped or fainting, then it generates an alert which is received and managed from a service center. DIA is framed within the second generation telecare systems, it solves the long waiting time which a person who is alone at home after suffering a fall or faint is often subject to. It eliminates the possibility that this incident goes unnoticed because the system is able to detect itself. The product created allows ubiquitous monitoring for home care of older people living alone, in a non-intrusive way and preserving their privacy.

The sensor network consists of a set of sensor nodes, to be distributed ubiquitously at home. The sensor nodes implement the minimum architecture to carry out environmental monitoring activities, develop messages with information about events that occurred and to form a wireless network with mesh topology over which the messages are taken to the PC. These nodes comprise a microcontroller and a radio transceiver specifically for low-power, as well as a set of commercial sensor devices able to pick up the activity at home in a non-intrusive and safe way. In particular, passive infra-red (PIR) sensors are used to monitor the location of the user, pressure-detecting mats under the mattress to detect the presence in bed and magnetic contact sensors to signal the opening of the front door of the house.

2.2 Bed/Seat Occupancy Detector Issues

Pressure mats have been appropriate as a first approach to detect the bed or seat occupancy. However, their functionality is dependent on various environmental factors such as the type of mattress and bed base or weight of the user. It's operation is simple. It consists of two foils separated by a layer of foam with cylindrical holes. Both sensor foils conform a normally-open contact (NO) that closes under the weight of the user. The main disadvantage of this device is that, once made, its sensitivity is determined by the diameter of the holes. The sensitivity required is dependent on the type of mattress, bed base and weight of the user. The use of mats with different sensitivity to each individual is not feasible on a large scale. For these reasons finding adaptive systems to maintain consumption and bounded cost it is desirable.

The use of sensors resistive based transducers could be an alternative, such as FlexiForce [7] from Tekscan Manufacturer. It consists of two sheets of polyester substrate. The transducer is modelled as a variable resistance dependant of pressure on the sheet. Researchers at Massey University in New Zealand have developed a bed occupancy sensor which generates a digital signal indicative of the presence by implementing a configurable threshold [5], [6]. This device may be suitable for AAL systems. However, the authors do not refer to the energy consumption of the implemented sensor and a quick inspection of the proposed circuit shows that consumption will always be greater than 2.5 mA. This makes this design is inappropriate for a network of wireless sensors powered by batteries, with a global average consumption (including sensing, data processing and communications) below 3 mA.

A third alternative is proposed in this paper, based on MEMS (*Micro-Electro-Mechanical Systems*). Specifically, EMFi (*Electro-Mechanical Film* [8]) based traducers can be used to implement flat capacitors whose capacity is dependant of the force exerted over it's surface. These devices, properly covered, may be placed under one leg of the bed to allow a rough measure of the exerted force. Sensor node would apply periodic signals filtered by a circuit tuned by the EMFi device and, allowing capacity and, therefore, the exerted weight measuring. The sensor node would be calibrated on place, before their operation. This solves the problem of lack of adaptability. Re-calibration routines could be also made using information from the context extraction and reasoning layers.

Table 1. EMFIT transducer properties

Property	Symbol	Value	Unit	Tolerance
Storage Temp.	Ts	-40 to 50	$^{\circ}C$	
Operation Temp.	Tr	-20 to 50	$^{\circ}C$	
Thickness	D	70	μm	
Sensitivity	Sq	25-250	pC/N	±5%
Op. Range	P	N/cm^2	> 100	

Table 2. EMFIT transducer characterization: Capacity Vs Static Force

Force	**EMFi 1**	**EMFi 2**		**Force**	**EMFi 1**	**EMFi 2**	
	Test 1	**Test 1**	**Test 2**		**Test 1**	**Test 1**	**Test 2**
$[kg]$	$[pF/cm^2]$	$[pF/cm^2]$	$[pF/cm^2]$	$[kg]$	$[pF/cm^2]$	$[pF/cm^2]$	$[pF/cm^2]$
0	65-68	61-62	60	25	86-87	67-69	
5	70-73	62-63		30	87-89	68-70	
10	75-77	64-66		35	90-92	69-71	
15	79-81	65-66		40	93-94	70-71	73
20	81-83	67-69	70	45	95-96	70-72	
				50	96-97	72-73	

2.3 EMFi Transducer Properties and Features

Electromechanical Film (EMFi) is a thin, cellular, biaxially oriented polypropylene film. High sensitivity, light weight and relatively low cost are the main advantages on EMFi. This transducer can be modelled as a charge source dependant of dynamical forces exerted on its surface as follows [9] [8]: $\Delta q = k_s \Delta F$, where k_s is the transducer sensibility. Main properties of the commercial model from manufacturer Emfit Ltd. [10] are shown at Table 1.

On the other hand, the transducer is a capacitor with variable capacity depending on the force exerted on its surface. Characterizing the transducer in this way is not present in the literature. We have conducted some basic tests, using a hydraulic press with digital pressure meter, and a capacimeter, in order to obtain a first characterization. The tests were conducted on samples from the S-Series EMFIT manufacturer in its circular format of 1 cm radius, see Figure 1.

The procedure is as follows: (1) A sample of EMFi is connected to the digital capacimeter terminals. (2) The sample is introduced into the hydraulic press and the weight is set to 0 kg, with the sample slightly trapped by it. (3) Finally, the pressure exerted by the press is increased in sections of 5 kg to reach 50 kg, registering the capacity offered by the transducer at each step. This test has been made with two different samples of the transducer. Besides, a second test has been made over one of the samples. The selected force range matches with the weight that would be exerted on every single leg of the bed or seat [0-50] kg, considering 4 as the standard number of legs and 200 kg as the maximum weight of a person. Table 2 shows the results of the tests. Every row contains the capacitance measures for every sample/test at a specific force. The capacitance has been noted as a range of registered values, since the reading was not completely stable. An approximately logarithmic relation has been observed.

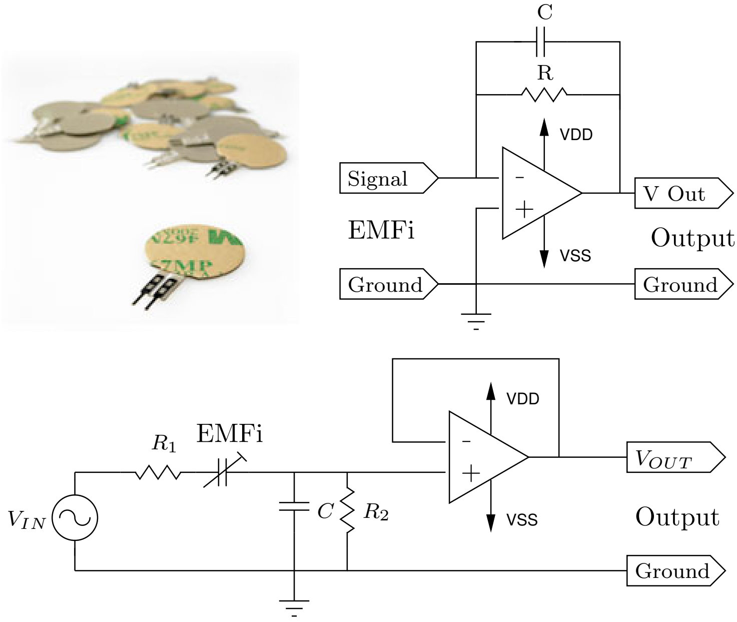


Fig. 1. On the left, EMFIT S-Series. On the right, Charge Amplifier for EMFIT. Below, voltage divider to get the EMFi transducer capacity.

3 Design and Implementation of EMFi Based Occupancy Sensor

3.1 Design of the Sensor Device

The low-power sensor device consists of two functional blocks: a quasi-passive force change detector and an active occupancy sensor, by means of weight measuring.

The force change detector design is based on the schematics proposed by the EMFIT manufacturer for dynamic forces measuring [11]. This schematics use the sensory model in which the transducer behave as a charge source which is dependant of the force applied between it's faces. The circuit shown in Figure 1 is a charge amplifier which transforms the signal transducer into a voltage signal. The output of this assembly is proportional to the variation of the pressure supported by the transducer. This signal is then threshold to generate interrupts to the micro-controller, thereby acting as a detector of changes in weight, with low power consumption.

The second block, the occupancy sensor, employs the EMFi transducer with it's model as a force dependant capacitor. Transducer's capacity is measured by means of a capacitive voltage divider. As shown in Figure 1, the voltage divider is feed with an AC signal when a measure is to be sampled. The resulting output is a signal with the same frequency but amplitude-modulated by the force on the transducer. This signal, after impedance conditioning, can be directly sampled and processed by a micro-controller to get a capacity estimation; which is also an estimation of the weight and the occupancy probability. A variant of the schematics is also proposed in which, capacitor C and EMFi transducer swap positions.

3.2 Implementation and Integration in the Wireless Sensor Node

The sensor node prototype incorporates a connector for an EMFi transducer. It's terminals are wired to an analog switch that allows alternative connections to either of the proposed circuits. It implements the described force change detector and weight sensor. The second one offers two variants, as mentioned, and the prototype has been built to allow testing of both, depending on the assembled components.

The force change detector output has been adapted to generate two different external interrupts to the micro-controller. The weight sensor output has been driven to the analogical-to-digital converter (ADC).

The application programmed on the micro-controller (firmware) manages the external interrupts from the force change detector, feeds the capacitor voltage divider with a square signal of 100 kHz and, finally, samples and processes it's output. The sequence is as follows: (1) When a force change occurs, the device fires a weight measuring over the voltage divider. (2) It compares the result to the previously calculated threshold and fires an event to communicate the bed/seat occupancy. (3) The event is transmitted via radio to the PC trough the base station. Additionally, a LED is lighted when occupancy detection is positive.

The signal is not static so the measurement is calculated from 150 samples from the ADC. Several algorithms have been tested to get an stable and significant value. After the test the following one has been selected: The first 10 values are used to get a starting average value. After that, every single sample (m_n) is added to the current average value $M_n = M_{n-1}(1 - \frac{c}{n}) + m_n \frac{c}{n}$, where c is a pondering coefficient, which is established by comparing the current sample with the previous average as follows: $c = 1$ when $\frac{m_n}{M_{n-1}} \in [0.9, 1.5]$. $c = \frac{1}{3}$ when $\frac{m_n}{M_{n-1}} \in [0.5, 0.9) \cup (1.5, 2]$. $c = 0$ otherwise.

4 Results

4.1 Results for the Occupancy Detection

The prototype of force changes detector has been assembled according to the schematics shown on Figure 1 with the following values: R=1 MΩ C=220 pF. This value for R is recommended by the manufacturer of the transducer, while the value of C is that for a time constant of $tau = \mathrm{RC} = 220\,\mu\mathrm{s}$; enough for the signal to be detected by the microcontroller. Figure 2 shows the ouput of the analogical stage. Signal remains at virtual ground ($Vcc/2$) when the transducer is not perturbed. When a force increasing happens (sitting fact), signal goes to 0 V while about 100 ms and goes back to its rest value. When force is decreased (standing up fact), signal is fired to Vcc and comes back to rest in the same way. This signal is digitalized by thresholds to get two signals that actuate as external interrupts to the micro-controller, as shown in Figure 2. Complex movements on the bed or couch, with different intensities, could lead to partial and unreliable detection, as the oscillations observed in Figure 2. This justifies the need of weight measurement on the system at rest to ensure the occupancy.

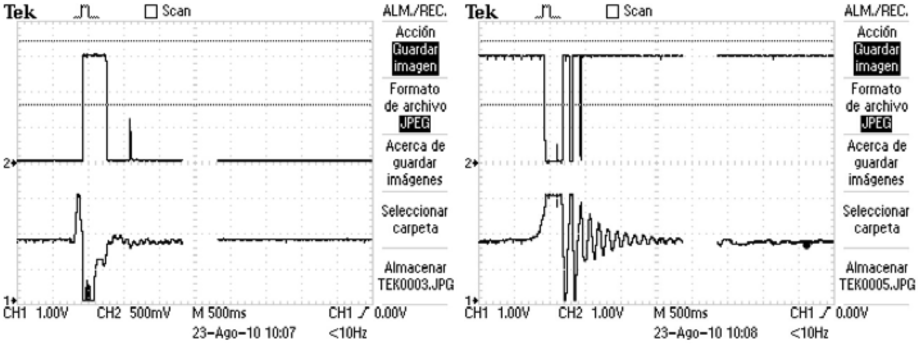


Fig. 2. Force change detector outputs. On the left, force increment detection (sitting fact); on the right, force decrement detection (standing up fact.)

Test have been effectuated on the occupancy sensor prototype for several values for the parameters pointed at Seccion 3. Resistor R_1 has been adjusted to suppress the over-shot (OS) keeping the rising time (T_r) as low as possible. Resulting values are: $R_1 = 150\,\Omega$, $OS = 0\%$ y $T_r = 70\,ns$.

Remaining hardware parameters, as well as the data processing algorithm, have been tuned by means of test series in the following way. The mote has been configured to sample and transmit the weight every 2 s. All combination of standard values in a range have been tested for both variants of the schematics with every processing algorithm among three proposed ones. The samples have been recorded for two states: (1) only a chair weight and (2) chair plus 85 kg subject. All the samples have been tabulated to find the configuration with the best stability on readings (lowest variation range on each state) and the highest distance between readings from one state to the other. The final configuration is as follows: $R_2 = 1M\Omega$ from the range $[100k\Omega - 10M\Omega]$ and $C_2 = 470pF$ from $[2pF - 680pF]$ with the schematics and processing algorithm described in Seccion 3.

The sensor cluster has been validated by people with different weights. The sensor has properly indicated the chair occupancy for every subject. Readings of the analog sensor and calculated thresholds are shown on Table 3.

Table 3. Final results for occupancy sensor with different weight people

Weight	ADC Readings		Calculated
$[kg]$	Empty	Occupied	Threshold
50	0x60	0x57	0x5C
85	0x60	0x59	0x5C
110	0x62	0x5A	0x5D

4.2 Power Consumption

The design of the new occupancy sensor for bed and couch will only be valid if the power consumption is low significant over the average node consumption, which results about 3 mA. Occupancy sensor consists of two different modules, the first one detects force changes while the second one effectuates weight readings. First module must remain permanently active while the second one is only operated after a force change detection.

Power consumption from the force change detection module is confined to the consumption from the operation amplifier plus two $100\,k\Omega$ resistors connecting V_{CC} and Gnd to generate logic level signals by means of MOS transistors commutation. Overall power consumption estimated for the force change detector is $260\,\mu A$, about 10% fo the node average consumption.

The weight meter has a higher consumption when active. However, weight measurements will only be triggered by detecting variations, so the average consumption gets down. The time spent for the micro-controller to get the 150 samples has been measured in $T_M = 74\,ms$. This implies a task duration of $T_{Check} = 5 \cdot T_M = 370\,ms$, with an average consumption of $I_{Check} = 10\,mA$, which is the micro-controller consumption. To estimate the average consumption of the device is necessary to estimate an approximate read rate: (1) It is considered to be detected no changes in weight, and therefore there will be no reading, in a vacant bed or chair. (2) In a conservative way, it is considered that an occupied bed or chair will present a mean period between detections of variation of $10\,s$. It is possible to implement a state machine to establish a maximum rate of measurement, assuming the consequent delay in the generation of events. (3) A person may occupy a bed or chair for an average of 8 hours per day. With these three assumptions, the duty cycle for the occupancy sensor is $DC = \frac{8\,h \cdot 370\,ms}{24\,h \cdot 10\,s} = 0.012\%$ and the average consumption generated by it would be $I = DC \times I_{Check} = 120 \;\; muA$, around 5% of the average consumption of the node.

5 Discussion and Conclusions

The goal of this work was to obtain an occupancy sensor for bed or couch that could improve the contact mats previously employed in the sensor network for a set of Ambient Assisted Living (AAL). This interest is due to the lack of adaptability that mats presented to various factors such as hardness and type of mattress and bed base, in the case of the bed, and the hardness of the structure in the case of the couch, as well as the weight of the person in the one case and the other. Other sensors based on resistive transducers do not meet the low power constraints. The proposed sensor is independent of these factors except the user's weight, however, the possibility of calibration makes this aspect does not become a problem. Occupancy sensor developed has proven effective in laboratory tests. It has been tested with people of different weights. On the one hand, detections have been correct. On the other hand, the recorded values have been stable and away enough from the threshold to consider that the sensor can behave properly in an actual installation. A fundamental aspect of the design is energy consumption. According to the estimations, the consumption of a node to incorporate occupancy sensor would be increased by 15%, which is acceptable for implementation.

Finally, future direction would point as redesigning the sensor so that it can be a stand - alone device that does not require a micro-controller to operate. An implementation of this could even be transferred to the field of microelectronic design for manufacturing as an integrated circuit, which would be translated into an especially significant reduction in cost and energy consumption.

Acknowledgment. This work is partially supported by Ambient Intelligence & Interaction S.L.L. (AMI2) and by the Spanish Ministry of Ciencia e Innovación (MICINN) under grant TIN2009-14372- C03-02.

References

1. Ambient Intelligence & Interaction SLL (Ami2) Company web site, http://www.ami2.net
2. Fernández-Luque, F., Zapata, J., Ruiz, R., Iborra, E.: A wireless sensor network for assisted living at home of elderly people. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5602, pp. 65–74. Springer, Heidelberg (2009)
3. Blaya, J.A.B., Palma, J., Villa, A., Perez, D., Iborra, E.: Ontology based approach to the detection of domestics problems for independent senior people. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5602, pp. 55–64. Springer, Heidelberg (2009)
4. Gaddam, A., Mukhopadhyay, S., Gupta, G.: Necessity of a bed-sensor in a smart digital home to care for elder-people. In: IEEE Sensors, October 2008, pp. 1340–1343 (2008)
5. Gaddam, A., Mukhopadhyay, S., Gupta, G.: Development of a bed sensor for an integrated digital home monitoring system. In: International Workshop on Medical Measurements and Applications, MeMeA 2008, May 2008, pp. 33–38 (2008)
6. Gaddam, A., Mukhopadhyay, S., Gupta, G.: Intelligent bed sensor system: Design, expermentation and results. In: IEEE Sensors Applications Symposium (SAS), February 2010, pp. 220–225 (2010)
7. Flexiforce® sensors - specifications and features. Technical report, Tescan, Inc., http://www.tekscan.com/flexiforce/specs-flexiforce.html
8. Lekkala, J., Paajanen, M.: Emfi-new electret material for sensors and actuators. In: Proceedings of 10th International Symposium on Electrets, ISE 2010, pp. 743–746 (1999)

9. Paajanen, M., Lekkala, J., Valimaki, H.: Electromechanical modeling and properties of the electret film emfi. IEEE Transactions on Dielectrics and Electrical Insulation 8(4), 629–636 (2001)
10. Emfit Ltd. Thin film ferro-electret sensors, http://www.emfit.com/en/sensors/products_sensors/
11. Emfit Ltd.Preamplifiers for emfit sensors. Technical report., http://www.emfit.com/uploads/pdf/Emfit_preamplifiers_for_emfit_sensors.pdf

Protocol Integration for Intelligent Monitoring Applications in Wireless Sensor Networks

Antonio M. Ortiz, Fernando Royo, Teresa Olivares, Luis Orozco-Barbosa,
José Carlos Castillo, and Antonio Fernández-Caballero

Albacete Research Institute of Informatics
University of Castilla-La Mancha, 02071 Albacete, Spain
{amanuel,froyo,teresa,lorozco,josecarlos,caballer}@dsi.uclm.es

Abstract. Integration of algorithms and protocols from different layers will make possible the deployment of large-scale wireless sensor networks which are the basis of interesting monitoring applications. The growing number of nodes that comprise within these networks requires a correct organization and efficient node synchronization to ensure data reliability. In this study, we focus on the integration of fuzzy-logic based routing with a TDMA MAC protocol. By considering the experimental results of them working separately, we have integrated them to work together. The use of a rapid configuration and efficient slot assignment from the MAC protocol, and the accuracy of the logical tree created using fuzzy logic, allows to have a network in which nodes are both organized and synchronized, while load balance extends network lifetime.

Keywords: Wireless sensor networks, Fuzzy-logic, MAC, Routing, Cross-Layer Integration.

1 Introduction

Wireless sensor networks (WSNs) comprise a large number of sensor devices that communicate with each other via wireless channels, with limitations of energy and computing capabilities. The efficient and robust realization of such large, highly-dynamic and complex networking environments is a challenging algorithmic and technological task [1].

Our research takes a cross-layer-based approach, working with the components of an emerging general-purpose sensor networking infrastructure. Networking is important because it provides the glue that allows individual nodes to collaborate. Radio communication is the major consumer of energy in small sensor nodes. Thus, the optimization of networking protocols can greatly extend the lifetime of the sensor network as a whole [2].

This paper focuses on both the link and network layers. The former layer is approached by considering a media-access control (MAC) protocol, while topology control and a routing approach are discussed for the latter. The MAC layer must manage the communication channel available for the node, and so must avoid collisions and errors in communications.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 511–520, 2011.

Organizing a network, composed in many cases of a high number of low-resourced nodes, is a difficult task since the algorithms and methods have to save as much energy as possible while offering good performance. Power-saving has been the main driving force behind the development of several protocols that have recently been introduced. In this context, perhaps the greatest energy savings are achieved by protocols whose communications are based on time division multiple access (TDMA) and synchronization. However, synchronous communications require the organization of the network nodes in an efficient structure such as a logical tree. This structure can be used to achieve both, collision-free communications and minimum end-to-end latency. The network layer is responsible for the dynamic-routing tree creation and topology control using a novel and interesting routing algorithm.

These kinds of algorithms take into account important node features such as battery level, number of hops to the sink or link quality; note that if this information is not considered, problems in the network may occur (interrupted paths, packet loss or isolated nodes, among others).

Our work integrates the SA-MAC protocol [3], a simple but effective collision resolution protocol as a means to set-up the slot allocations of TDMA protocols, with NORIA (Network rOle-based Routing Intelligent Algorithm) [4], a novel routing algorithm for wireless sensor networks. In this paper, we mainly focus on how SA-MAC and NORIA work together in dense networks in order to optimize the protocols that control the behaviour of network nodes in real applications. In our group, we have developed a multisensory indoor monitoring application, oriented to energy management in intelligent buildings, where we have encountered a number of problems related to information loss and packet collisions. Due to this, we have decided to optimize network and medium access protocols and integrate them together in order to improve the overall performance.

The rest of this paper is organized as follows: Section 2 summarizes some related work. The synchronized medium access protocol, SA-MAC is detailed in Section 3. Fuzzy logic principles are presented in Section 4, and serve as a basis to the intelligent routing approach detailed in Section 5. Section 6 presents the integration of both approaches and details the experiments and obtained results. Finally, Section 7 gives some concluding remarks.

2 Previous Work

TDMA MAC protocols avoid some important problems such us idle listening, collisions, overhearing and overhead due to control packets. In addition, when an efficient global synchronization mechanism is available, the topology is fairly stable and TDMA protocols are usually the best option for efficient data communications in WSN [5]. A few relevant MAC protocols that make use of TDMA channel sharing are worth mentioning in [3]. SA-MAC protocol solves this set-up phase successfully and performs a second phase of synchronization and data transmission. Also, it is strongly recommended for a MAC protocol to be

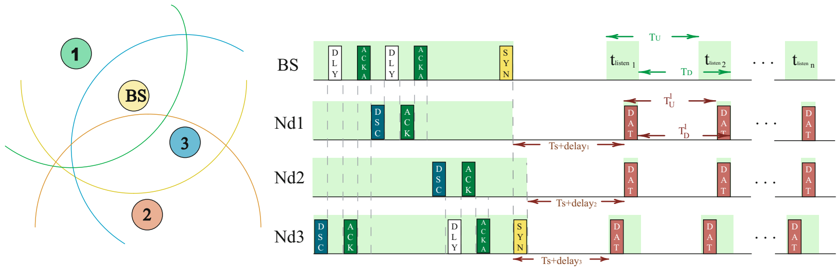


Fig. 1. SA-MAC timing diagram

distributed and self-organized to support topological changes, since these features are essential to ensure the efficient scalability of a wireless sensor network [6]. In order to do this, we integrate SA-MAC with NORIA, which will create the logical tree in an optimal and intelligent manner, considering the differences between nodes and always choosing the most adequate ones to be included in the routes.

In [7] and [8] different routing and self-organizing techniques for WSN are described. These techniques serve as a basis for future routing algorithms which are intended to make the network more efficient. After all, we have came up with the idea of using artificial intelligence techniques to help the decision-making process in order to get more efficient algorithms. Artificial intelligence techniques reinforce the efficiency and the performance of routing algorithms, by combining data from nodes and their interactions in order to make decisions oriented to improve global network performance. Decisions related to information transmission from a source to the sink are one of the most important aspects in sensor networks. Our approach tries to show how artificial intelligence techniques, specifically fuzzy logic, can help these decision-making processes to improve efficiency and to extend network lifetime. We are very interested in studies combining fuzzy rules with networking. In [9] the authors propose two asynchronous MAC protocols, they make a complicated schedule interval and they design a rescheduling fuzzy logic system to monitor the influence of accumulative clock-drifts, the variance of traffic strength and service capability on communications. In their experiments, they increase the number of nodes within a cluster from 5 to 30 and exclude the factors coming from physical layer and network layer to simplify the analysis. Another interesting approach is taken in [10], where the authors propose FuzzyMAC, a CSMA/CA-based MAC protocol that utilizes two separate fuzzy logic controllers to optimize both the MAC parameters and a sleeping schedule duty-cycle. The experiments only show results for 50 nodes and use a proprietary simulator.

3 Synchronized MAC

Power saving has been the main driving force behind the development of several protocols that have been recently appeared in the literature (see [11] for a recent survey). In this context, the largest energy savings are achieved by protocols whose communications are based on time-division multiple access (TDMA) and synchronization.

3.1 SA-MAC Protocol Overview

The main aim of the SA-MAC protocol is to synchronize the ON/OFF periods of senders and receivers. In the following, the protocol operation will be described by considering a network consisting of a sink node (or base station) responsible for gathering all the data sensed by all the other nodes. Some of the other nodes may have, when required, to act as relays enabling the forwarding of the collected data to the sink station.

The operation of SA-MAC is divided into two phases: 1) the set-up phase and, 2) the synchronization and data transmission phase. In this paper we will briefly describe both of these. Other aspects of the protocol operation can be consulted in [3].

During the set-up phase of the SA-MAC protocol the network nodes exchange four types of packets, namely discovery packets (DSC), delay packets (DLY) and acknowledgement packets (ACK and ACKASOC). In the simplest scenario the set-up phase starts when the base station (e.g., the sink node) announces its presence as a father node so that all other nodes can start trying to establish a father-and-child relation. Thus, all nodes that become aware of the presence of the base station start to broadcast discovery packets (DSC). Upon receiving a DSC packet, the base station sends a DLY packet to the corresponding node.

The delay packet indicates the time slot that is assigned for transmissions from the sensor node to the base station. The node acknowledges the DLY packet with an acknowledgement packet, ACK, and this acknowledgement packet will be replied to another one from the parent node named ACKASOC. In this way, the sensor node finishes its association to the base station and then it may become a father node for other nodes.

In order to illustrate the operation of the SA-MAC protocol in a more complex scenario, consider a set of nodes consisting of a base station (BS) and nodes Nd1, Nd2 and Nd3. Let us assume that Nd1 and Nd3 are located within the coverage area of the BS and that Nd2 is located within the coverage area Nd3, but out of reach of the BS. Once the BS announces its presence, nodes Nd1 and Nd3 can start sending DSC packets and collisions may occur at this time. Let us assume that Nd3 resolves the collision and sends its DSC packet to the base-station and establishes a father-and-child relation with the BS. At this point in time, Nd2 chooses the Nd3 node as father because it is the nearest one to the BS and if Nd2 detects that the channel is free, it can start to send its DSC packets. Nd1 detects the end of the association between Nd3 and BS and it sends its DSC, establishing a father-and-child relation too. Once all the nodes in the first routing level have been associated, nodes from the second one can start to send their DSC packets. When choosing the best father, the nodes of a lower level listen the channel to catch the information from the ACK and ACKASOC packets from the nodes of the upper level. These packets include information about the number of hops to the BS; actually, only this parameter has been taken into consideration in the initial design. Once the network has been created, the BS waits for a period of time, during which no new nodes are detected. The BS node will then transmit a SYN packet indicating the beginning of the data transmission phase. This process starts at the base station and a SYN packet is propagated among successive levels. Figure 1 shows the sequence of actions described for this scenario.

During the second phase, the nodes will turn ON and OFF, thus completing a duty cycle. This is to say, the parent node will wake up and each node will transmit within the slot allocated during the set-up phase. In this way, we limit the overhead due to the transient period every time the radio interface is turned on.

4 Fuzzy Logic

Fuzzy logic is a decision system approach which works similarly to human control logic. It provides a simple way to arrive at a definite conclusion based upon imprecise, vague or ambiguous input information. Furthermore, only a few data samples are required in order to extract the final accurate result. Besides, fuzzy logic is a handy technique since it uses human language to describe inputs and outputs [12]. All these features make fuzzy logic appropriate for the parent decision process in wireless sensor networks.

One of the frequently-used fuzzy-logic inference methods is Mamdani [13], which consists of four phases: fuzzification, rule evaluation, combination or aggregation of rules, and deffuzification. The input of a Mamdani fuzzy-logic system is usually a crisp value. To allow this value to be processed by the system, it has to be converted to natural language, that is, fuzzified. In this way, the fuzzifier takes numeric values and converts them into fuzzy values which can be processed by the inference system. These fuzzy values represent the membership values of the input variables to the fuzzy sets. Once values have been fuzzified, the inference system processes the fuzzy rules to get a fuzzy output. The third step in the Mamdani inference method is the aggregation of all outputs, where the outputs of each rule are combined to form a new fuzzy set. Finally, in the deffuzification process, the new aggregated fuzzy set is converted to a number. In summary, fuzzy logic is an efficient method for making decisions and it also has the advantage of working with natural language and require low computational capabilities.

5 Network Routing

Routing is a key element in the operation of wireless sensor networks. Paths from any node in the network to the base station must be defined in order to get efficient network communication. With the use of fuzzy-logic, we expect to get successful results in terms of network organization time and number of hops from any node to the base station, always choosing the best parent in terms of energy and number of hops to the sink.

5.1 Fuzzy-Logic-Based Routing

NORIA is an algorithm capable of route data in a network by creating a communication tree. The process starts at the base station and it is then propagated hop by hop until covering all the network nodes. Node conditions are evaluated by using a fuzzy-logic-based mechanism, and only those nodes with better state (battery, number of hops to the sink) act as data forwarders (parent nodes).

The algorithm establishes minimum paths, in terms of energy spending, from every node to the base station, which gathers data coming from all the network nodes. Roles are assigned to network nodes in order to balance the network load. These roles are *leaf*: nodes that sense the environment and send data to the base station, and *master*: nodes that as well as performing *leaf* node tasks, also forward data from other nodes to the base station.

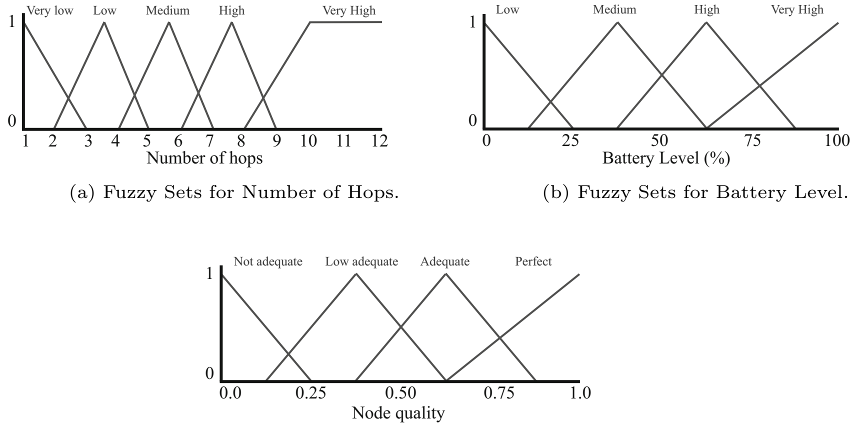
In order to perform network organization through parent election and role assignment, NORIA stores neighboring data in a neighbor table, and uses the output of a fuzzy-logic system as a metric to make decisions.

NORIA defines four kinds of messages in order to organize the network and send collected data. These messages are: IPM (Information Propagation Message), which includes local information; RDM (Role Decision Message), which informs about selected role, MRM (Master Request Message), to request neighbor role changing, and DM (Data Message), which includes requested and/or forwarded data.

The algorithm consists of two stages: first, the algorithm organizes the network by creating the communication tree, which is routed at the base station, and by fixing as forwarder master nodes those with better conditions. The second stage is still being developed, and performs periodical verification, which includes, if necessary, role migration in order to provide load balancing and to avoid node overloading.

In the first stage of the algorithm, the communication tree is created and roles are assigned to nodes. The organization procedure starts when the base station sends an RDM. Nodes that receive this message send an IPM and start a timer. In this period of time, they wait for information messages from neighboring nodes. When the timer has expired, nodes perform parent election and make the role decision. An MRM is sent in case of the elected parent having a *leaf* role. Once the role has been decided, the nodes send an RDM to induce next-hop neighbors to start the organization phase.

The input variables to be considered in this experiments are: the number of hops to reach the base station, and the remaining energy. This parameters are an example of the full set of parameters which can be also included in the decision process (delivery probability, delay, signal strength... among others). The output variable represents the suitability of the node to be selected as parent node. Figure 2 shows fuzzy sets for input and output parameters.



(a) Fuzzy Sets for Number of Hops.

(b) Fuzzy Sets for Battery Level.

(c) Output Fuzzy Sets.

Fig. 2. Input Fuzzy sets

Nodes will compare the output of the evaluation for each neighbor node to perform the parent selection process. Note that fuzzy input and output sets can be customized depending on the application and on the circumstances of each particular WSN. For example, in a network which needs real time data, it will be interesting the use of the end-to-end delay as a decision parameter.

For that example, the fuzzy rule base includes rules such us the following: if the *Number of hops* is *Low* and the *Battery Level* is *High* then the node quality is *Adequate*. Here, since we have 4 fuzzy sets for *Battery level* input and 5 for *Number of hops* input, we therefore have 20 rules, which are summarised in Table 1.

6 Combining Synchronized MAC with Intelligent Routing

The integration of a synchronous MAC protocol and an intelligent routing approach will make the network more accurate and efficient. Since SA-MAC uses just the number of hops of a neighbor to be elected as parent, integration with an efficient parent decision mechanism will be beneficial in terms of network performance and network load balance, avoiding forwarder-node overload and improving network lifetime. SA-MAC sends to NORIA the number of hops and the battery level of the neighbor node, and NORIA's fuzzy-logic-based decision process provides the node quality value. Then, the node with the best quality value will be chosen as parent.

6.1 Performance Evaluation

In order to evaluate the performance of the combined proposal, we have implemented both SA-MAC and NORIA in Omnet++ [14] simulator. We are going

Table 1. Fuzzy rule base

N. of hops	Bat. Level	Node quality
Very low	Low	Low adequate
Medium	High	Adequate
Very low	Medium	Low adequate
Medium	Very high	Adequate
Very low	High	Adequate
High	Low	Not adequate
Very low	Very High	Perfect
High	Medium	Low adequate
Low	Low	Not adequate
High	High	Low adequate
Low	Medium	Low adequate
High	Very high	Adequate
Low	High	Adequate
Very high	Low	Not adequate
Low	Very high	Perfect
Very high	Medium	Not adequate
Medium	Low	Not adequate
Very high	High	Low adequate
Medium	Medium	Low adequate
Very high	Very High	Adequate

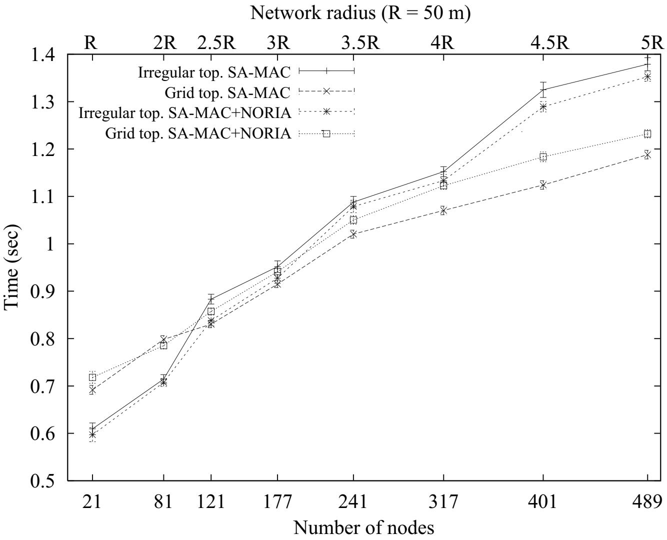
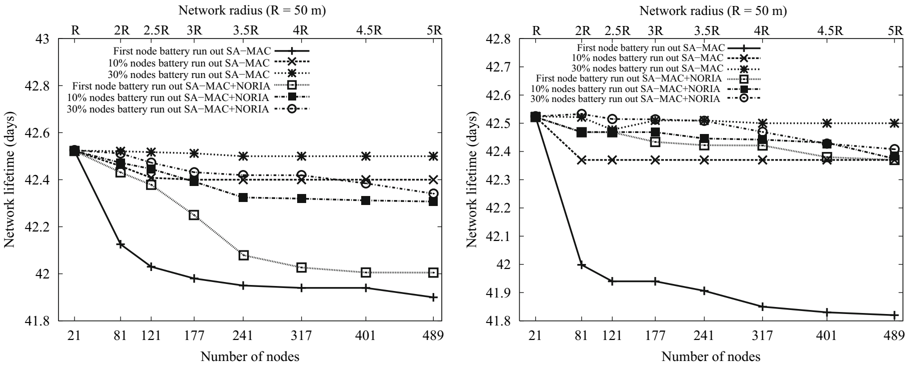


Fig. 3. Network set-up time



(a) Irregular topology.

(b) Grid topology.

Fig. 4. Network lifetime

to consider circular network areas with constant node density. The network size varies from 50 to 250m, and the number of nodes varies from 21 to 489. We have used both grid and irregular topologies for the purpose of experimenting with different node deployments. The experiments have been executed 100 times in order to obtain reliable results.

Figure 3 shows that the addition of fuzzy logic only increments the set-up time when we are working with the grid topology, due to the fact that the nodes are frequently changing for a best parent, compared to the original SA-MAC way of working.

The lifetime of the network, in general terms, is improved. For both cases, grid and irregular, (see Fig. 4), the time in which the first node's battery run out is improved, and the gap between this event and 10% and 30% of the network lifetime is reduced, indicating that network workload has been balanced correctly in the continuous process of network re-configuration. What is noteworthy is the improvement on grid topology. Although the set-up time was slightly longer, this increase is not significant compared to the balance of workload among the nodes, and so the network lifetime significantly increased.

7 Conclusions

In this paper, we have integrated an existing TDMA-based MAC protocol with NORIA, an intelligent routing approach for WSNs.

Since the correct operation of a wireless sensor network requires that all nodes have to know who they have to send to, and at which moment, it is desirable that the network is both self-organized and synchronized. First of all, we have performed a deep study of previous work, and SA-MAC was selected to be integrated with a fuzzy-logic-based routing algorithm, NORIA.

Experimental results have shown that the combination of SA-MAC and NORIA reduces the depth of the communication tree, and that the network load has been balanced among nodes,thus increasing network lifetime.

These favourable results obtained in the simulator will be tested in a real application with the aim of confirm the improvements that the protocol integration provides.

Acknowledgement

This work was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046, TIN2009-14475-C04 and Ecosense II project under grant POII11-0118-3014.

References

1. Boukerche, A.: Algorithms and Protocols for wireless sensor networks. Wiley, Chichester (2009)
2. Hristu-Varsakelis, D., Levine, W.S.: Handbook of Networked and Embedded Control Systems. Springer, Heidelberg (2004)
3. Royo, F., Olivares, T., Orozco-Barbosa, L.: A synchronous engine for wireless sensor networks. Telecommunication Systems (2009)
4. Ortiz, A.M., Olivares, T., Castillo, J.C., Orozco, L., Marron, P.J., Royo, F.: Intelligent Role-Based Routing for Dense Wireless Sensor Networks. In: WMNC 2010 (2010)
5. Macedo, M., Grilo, A., Nunes, M.: Distributed latency-energy minimization and interference avoidance in TDMA wireless sensor networks. Computer Networks (2008)
6. Liao, W., Wang, H.: An asynchronous MAC protocol for wireless sensor network. Journal of Network and Computer Applications (2008)
7. Al-Karaki, J.N., Kamal, A.E.: Routing Techniques in Wireless Sensor Networks: A survey. IEEE Wireless Communications (2004)
8. Dressler, F.: A study of self-organization mechanisms in ad-hoc and sensor networks. Computer Communications Magazine (2008)
9. Ren, Q., Liang, Q.: Energy-efficient Medium Access Control Protocols for Wireless Sensor Networks. Eurasip Journal on Wireless Communication and Networking (2006)
10. Wallace, J., Pesch, D., Rea, S., Irvine, J.: Fuzzy Logic optimization of MAC parameters and Sleeping duty-cycles in wireless sensor networks. In: IEEE VTC (2005)
11. Bachir, A., Dohler, M., Watteyne, T., Leung, K.K.: MAC essentials for wireless sensor networks. IEEE Communications Surveys and Tutorials (2010)
12. Reznik, L.: Fuzzy Controllers. Newnes Publishing, Oxford (1997)
13. Jang, J.-S.R., Sun, C.-T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Englewood Cliffs (1997)
14. Omnet++ Simulator page, http://www.omnetpp.org/

Event Detection and Fusion Model for Semantic Interpretation of Monitored Scenarios within ASIMS Architecture

Ángel Rivas-Casado and Rafael Martínez-Tomás

Dpto. Inteligencia Artificial. ETSI Informática, Universidad Nacional de Educación a Distancia, Juan del Rosal 16, 28040 Madrid, Spain
{arivas,rmtomas}@dia.uned.es

Abstract. Semantic interpretation of monitored scenes implies the well-known problem of linking physical signals received by sensors with their meaning for a human. Our line of work aims to develop a global architecture, which we call "Architecture for Semantic Interpretation of Monitored Scenarios (ASIMS)", to integrate all the information processing abstraction levels, from the sensory agent process and coherent fusion of agent results to behaviour and situation identification. This work presents a specific structure for the acquisition and fusion of events from the object level, which forms part of the global ASIMS structure. In remote processing nodes, events caused by variations in magnitudes in the common data model are identified via finite automata models. These events are merged by the central node to solve the usual problems of centralised systems: synchronisation, redundancy, contradiction and heterogeneity of the information that they receive from different sources. For this we have broken down the fusion mechanism into three stages: Synchronisation, Standardisation and Fusion, which are described in the article with simple examples.

1 Introduction

The overall problem that we wish to address is that of interpreting the activities in the scene being monitored. Of course, it is the old problem of associating sensor signals with semantics, with a description similar to human comprehension, like a human understands or identifies the scene. Possibly the most generic term is that of "interpretation": a message is interpreted, a cardiogram is interpreted, a video is interpreted, etc. We specifically want to interpret what happens in a monitored scenario, identify events and activities. For this sensors are situated and their signals are processed in the hope of finding spatial and time data patterns or associations. These associations make it possible to ascend step by step in successive abstraction levels to reach conclusions connected with decision-taking: a situation has been identified that requires a coherent response, either by a security guard or doctor, an alarm centre, an automated emergency system, etc.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 521–530, 2011.

Furthermore, we aim for a multisensor system to diagnose a complex scenario, which will require using all the different sensor capacities in a co-operative/collaborative way, which will include mechanisms to interpret this information together, and will make it possible to identify behaviours and situations by merging data from the different sensors in the different stages of the system (INT3). We speak of diagnosing as a complementary stage to monitoring, more usually located in immediate detection of the lower abstraction level in interpretation [1].

Of course, in video-controlled environments, the camera sensor is of major importance, although other types of sensors have processing stages very similar to those of artificial vision (acquisition, segmentation, identification, tracking), but what is presented in this work is generalisable to inputs from all sensor types. Multiple sensor integration depends on the application and type of data or signals used for the specific application. It is fundamental to bear in mind the technical processing requirements of all and each of the video sources present in the environment under surveillance. The computation requirements may increase in large scenarios under surveillance: multiple cameras, visual and lighting difficulties, variety of sensors, etc. For this reason, strictly centralised processing is not posed as the solution to the problem because it does not allow great scalability. Conversely, embedded, distributed processing in nodes (that include data capture) seems to be a possible solution to this problem and it has usually been addressed with agents. [2]

A distributed video-monitoring system presents additional problems when integrating the information from each of the monitoring nodes. On this point we have redundant, contradictory and heterogeneous information, which is another disadvantage of the approach. The solution to the problem calls for filtering, merging and standardising the information from each observation and processing node in the monitored environment. Here is the focal point of this work, to provide a solution model.

Usually, to do a symbolic description of the scene from the video sequences, movement is analysed [3,4]. The works by [5,6,7] focus on recognising events related to vehicles and humans in the airport environment. In the continuation of these works [8,9] a method is presented to recognise video events using a tracking framework and Bayesian networks that use data on the surroundings and trajectory. In [10] the composed events are analysed with hidden Markov models. An activity is considered a composition of different "action threads", which are processed separately but related by specific time restrictions.

In our earlier works in this area, we started from an architecture true to the model for abstraction layers [1], which communicate with each other bottom-up (emergency) and top-down (feedback) (neurocumputing) in distributed multi-sensor capture and processing nodes (CEDI) that communicate with each other via events with a central nodel [11]. This central node processes the highest abstraction levels with declarative models [2,12], where an overall, more complete view of the monitored scene is required. Our line of work aims to develop a global architecture, which we call "Architecture for Semantic Interpretation of

Monitored Scenarios (ASIMS)", to integrate all the abstraction levels of the information processing, from the sensor agent process and coherent fusion of agent results to behaviour and situation identification. The architecture is currently in the development and prototype stage.

This article first gives a general description of the architecture (section 2). Then it describes the specific proposal for detection (section 3), based on finite automata models, and event fusion (section 4) in the intermediate abstraction levels, and examples of use are given. The work finishes with the conclusions and, in particular, with the approach for the future stages of the work.

2 ASIMS Architecture

Every automated interpretation system of monitored scenarios has a basic structure consisting of acquisition stages and information processing using abstraction and decision-taking levels [1]. In our system, acquisition and part of the information processing is done locally. The other information processing and decision-taking is done centrally. In the system, the processing is distributed between Remote Nodes and the Central Node. The Remote Nodes capture and process the information in the environment around them. They send this information to the Central Node, which is in charge of merging and processing it, and its results are then passed to the decision-taking system (Figure 1).

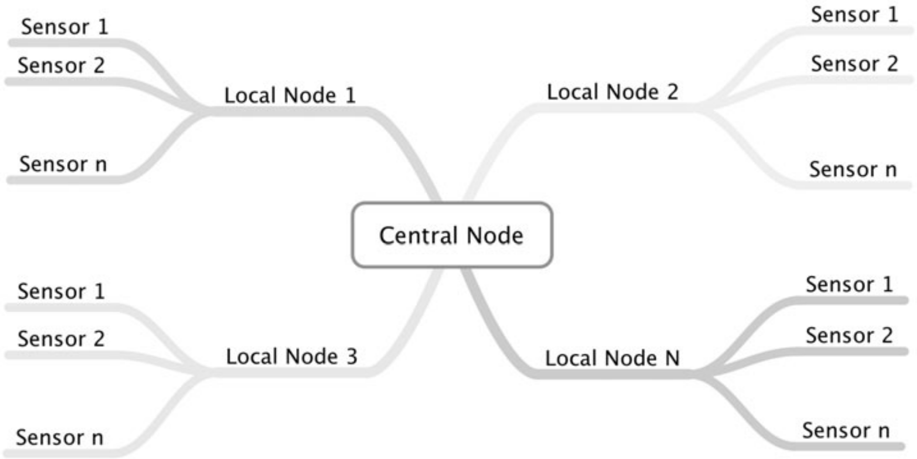


Fig. 1. Image of sensors and Local and Global nodes structure

Each Local Node has a common storage system at all levels "Common Data Model". All the levels of the architecture (acquisition, segmentation, identification, tracking, event classification and detection) use this (Figure 2). The levels are conceived as interchangeable modules within the architecture, which enable modules to be easily changed at any moment.

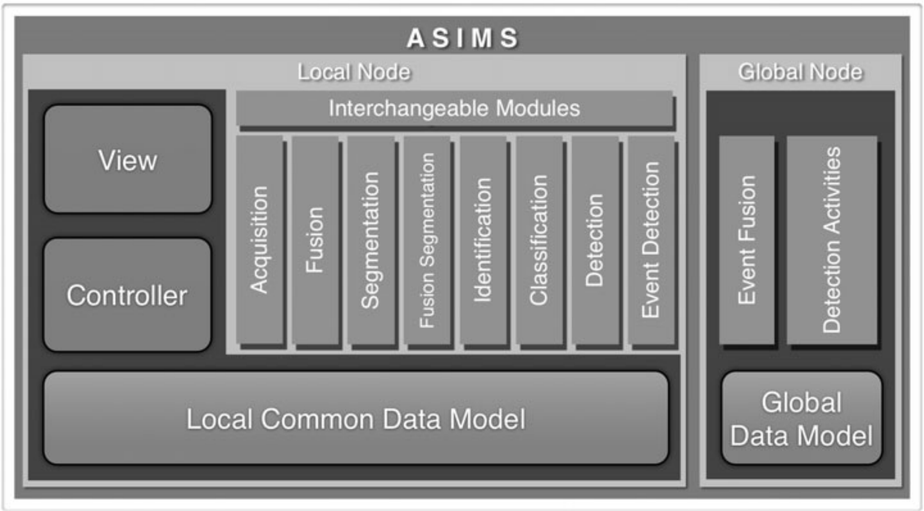


Fig. 2. Event Detection, Fusion and Activity Detection

The Common Data Model stores sensor information and the results from executing each of the algorithms: segmented images, number of objects in the scene, global position, speed, direction, shape, colour, class to which they belong, properties, detected events, etc. There is also a time marker or timestamp to label the information captured. All the modules have reading access, but they only allow writing in those data that they generate as output. The Central Node has two differentiated stages (Figure 2): Event Fusion and Activity Detection. The first of these is in charge of filtering information from all the Local Agents. The second detects global activities and generates the necessary "warnings" in pre-defined situations requiring human decisions.

3 Event Detection

In the description of the architecture in the previous chapter, Event Detection is in the last level of the Remote Nodes. The aim of this module is to identify patterns existing in the data structure of the Common Data Model, described in the previous chapter, in order to detect patterns that are produced in the monitored data. The characterisation of these patterns (that generate events) can be represented with finite automata for their resolution. Each of the events to be detected has its own associated finite automaton containing the necessary states to detect an event with its transition state. Thus, if we have three events to identify and two objects in the scene, three states will be generated by each Object:

$$Object1[Event1 = q0; Event2 = q0; Event3 = q0]$$
$$Object2[Event1 = q0; Event2 = q0; Event3 = q0]$$

As the transition conditions are fulfilled, the states of each event associated with each Object will make transitions. As a practical example, the event BeginToWalk is identified with the automaton in figure (Figure 3) . "q0" is the initial state where the automaton is when the event is generated and a transition can be made to state "q1" as long as the Object in the scene is a "Human", or to state "q3" if the Object in the scene is not "Human". The latter is the final state when the execution of the automaton finishes. From state "q1" a transition can be made to state "q2" if the speed of the Object in the scene is equal to 0. However, while the speed of the Object is greater than 0, the automaton will still be at state "q1". From states "q1" and "q2" a transition can be made to the final state "q3" if the Object disappears from the list of Objects in the scenario. From state "q2" a transition can be made to state "q1" if the speed of the Object is greater than 0 and at this very instant the event BeginToWalk associated with this Object is generated. If the speed is still 0 the automaton does not make a transition and stays at "q2".

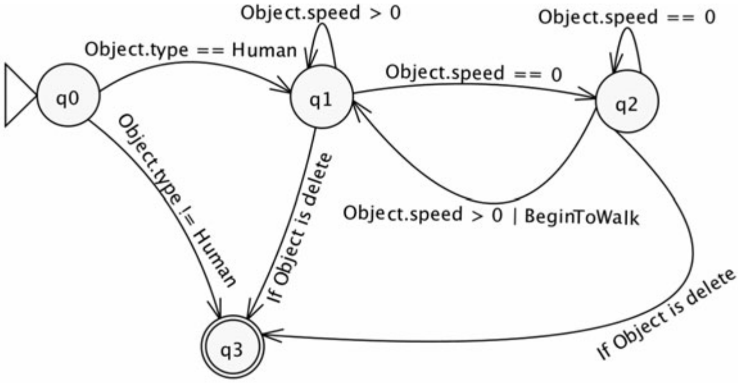


Fig. 3. Autómata para la deteccion del evento BeginToWalk

Similarly, all the events that can be generated are defined in the event detection module. The implementation of this method is immediate with verifications like "case" and "if", which provides a rapid, sequential, loopless execution. The automaton conversion to the source code is trivial, which makes it possible to work with automata diagrams with their resulting benefit for comprehension and analysis.

4 Event Fusion

The problems that centralised systems face are synchronisation, redundancy, contradiction and heterogeneity of the information from different sources. To address these problems it is necessary to apply methods that solve each of these stages. Thus, we break down the fusion mechanism into three stages: Acquisition, Fusion and Standardisation.

Synchronisation: It is in charge of asynchronous information retrieval from different sources. One of the principal problems lies in synchronising asynchronous information. For this the mechanism known as “time window” is used which establishes a time margin for retrieving the information before it is processed. In these time windows “time baskets” are in turn established where the labelled Events are stored with the same timestamp. A time window may have two or more time baskets. Both the time window time space and the number of time baskets are determined by the system requirements. Figure 4 shows an example of the time windows N and N+1. In the time window N, two time baskets are established for timestamps t and t+1. The events that arrive in the time window N are automatically classified into each of the time baskets with the timestamp that each of the received events has. When the time window N finishes, the time window N+1 is created and the timestamp t time basket is processed.

This process is necessary because the events with different timestamps are interspersed when they are received. This is due to communication and time delay factors in the information processing in each of the Remote Nodes. For example, the system can receive the following sequence of events in the time window N:

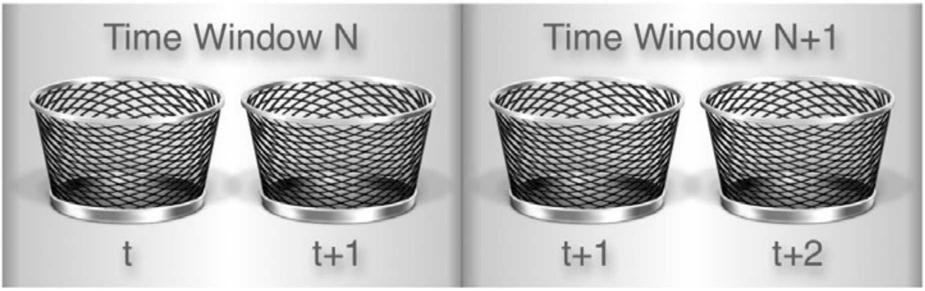


Fig. 4. Time windows and time baskets

The time space of the time window has to be wide enough to retrieve all the events with the same timestamp. As shown in the Formula (formula) this process is done considering the maximum processing time of a Remote Node, the communication delay and adding a security margin. This mechanism guarantees the reception of all the events with the same timestamp before they are processed.

$$\begin{aligned} TimeWindow(ms) = {} & MaximumProcessingTimeOfARemoteNode(ms) + \\ & EstimatedMaximumComunicationDelay(ms) + \\ & SecurityMargin(ms) \end{aligned}$$

Standardisation: The identification process of each of the Remote Nodes provides different labels for each of the scene objects. This generates inconsistent names for the same objects perceived by different Remote Nodes. To solve this problem names have to be standardised for the whole system. Using a method based on the global positioning of identified objects, equivalence tables are established (Table 1) in accordance with the Remote Node sending this information. With these tables and a translation service for names, the names for each scene object are standardised.

Fusion: In centralised systems with different clients sending information, contradictory or redundant information can be received at the same instant of time. This stage focuses on solving redundancies and contradictions from the different Remote Nodes. For this a model of the scenario is used where the information on the positions of the Remote Nodes and all the scenario objects is. To identify and solve redundancies the Events are grouped with the same timestamp according to their spatial reference in order to facilitate the process. Once the classification has been done a search is done for equivalent events. This equivalence is established in accordance with the Remote Sensor Nodes and the event equivalence formulation.

After the fusion, the resulting events are transmitted to the activity detection system (next processing level, which composes-abstracts events with greater semantic content) like that described in the article by R. Martinez-Tomas & A. Rivas-Casado [11]. This system is in charge of identifying spatial-temporal activities in the controlled environment in order to provide semantic information to the sequences of events that are perceived in the capture process.

The figure 5 represents a possible fusion situation where three sensors are monitoring a common area. Each Remote Node Event Detection Module identifies the event “At” at the same instant of time. The structure of the event “At consists of the set of attributes: human label ?h, position x ?x, position ?y and the instant of time ?t when the event is generated:

$$At((human\,?h)(x\,?x)(y\,?y)(time\,?t))$$

In the acquisition stage, all the names of the objects are extracted and compared in the table of existing names. Each remote node and name is assigned its global object name. This is done by positioning each object in the global world. In our example the acquisition stage receives three events “At” from three remote nodes (Node 1, Node 2 and Node 3) (Table 1)

Table 1. Name equivalence table

Remote Node	Object name	GlobalObject
Node 1	Human59	GlobalHuman65
Node 2	Human48	GlobalHuman65
Node 3	Human73	GlobalHuman65

The fusion of two or more events “At” generates a new event “At with the standardised name of the object, the mean of the positions x of all the events “At”, the mean of the positions and all the events “At” and the timestamp. The table 2 shows the group of events from the three nodes and how they merge to give a central event.

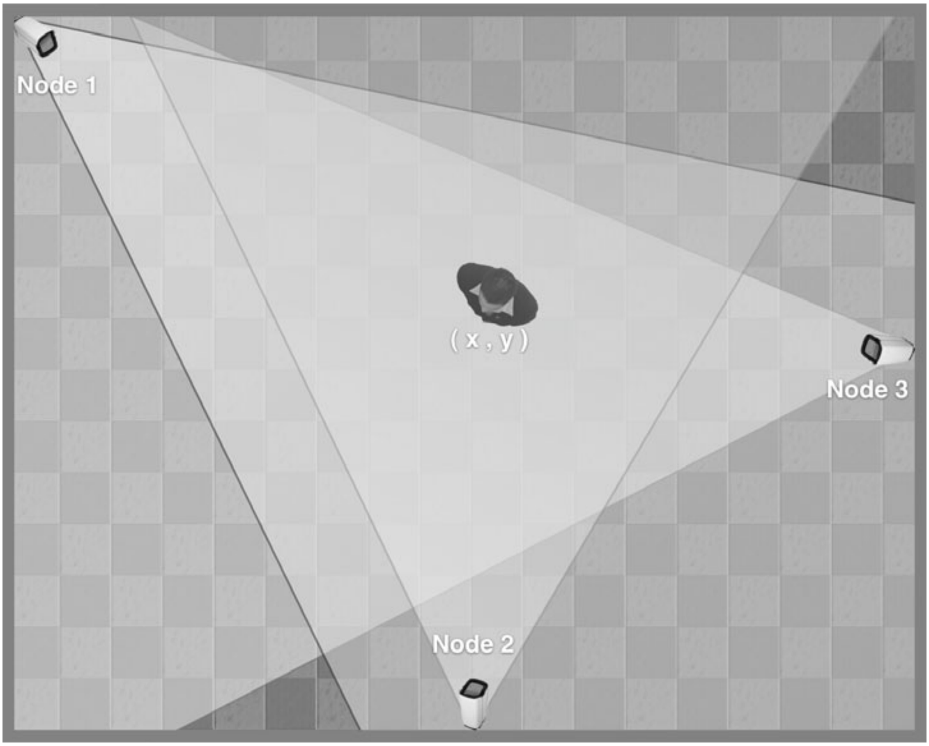


Fig. 5. Scenario model

Table 2. Event group table

Node	Type	Event	H	X	Y	Timestamp
Node 1	Acquisition	At	Human59	548	659	346543775
Node 2	Acquisition	At	Human48	547	657	346543775
Node 3	Acquisition	At	Human73	551	660	346543775
Central	Fusion	At	GlobalHuman65	548	659	346543775

5 Conclusions

The semantic interpretation of monitored scenes implies linking physical signals received by sensors with their meaning (ref al proyecto) for a human (semantic gap), i.e. with the diagnosis of what is occurring in front of him (if he were observing all of it), either as a support system or to activate a coherent automatic response system (without the human).

This work has presented a specific structure for the acquisition and fusion of events, from the object level, which forms part of the global ASIMS architecture. In remote processing nodes, events caused by variations in magnitudes in the common data model are identified via finite automata models. These events are merged by the central node to solve the usual problems of centralised systems: synchronisation, redundancy, contradiction and heterogeneity of the information from different sources. For this we have broken down the fusion mechanism into three stages: Synchronisation, Standardisation and Fusion, which have been described in the article with simple examples.

Continuing with the structure of abstraction levels, after identifying and merging events, we aim to continue with the architecture and its successive levels: identifying activities from the event composition and further identifying behaviours from the activity compositions. For this we aim to integrate the developments already shown in Iwinac-09 [11] into the highest-level central inference system and adapt them to the common ASIMS architecture.

Acknowledgements

The authors are grateful to the CiCYT for financial aid on project TIN-2010-20845-C03-02.

References

1. Martínez-Tomás, R., Rincón-Zamorano, M., Bachiller-Mayoral, M., Mira-Mira, J.: On the correspondence between objects and events for the diagnosis of situations in visual surveillance tasks. Pattern Recognition Letters (2007), doi:10.1016/j.patrec.2007.10.020
2. Casado, Á.R., Tomás, R.M.: Sistema multiagente integrado para el apoyo de la video identificación en entornos de seguridad controlada. In: Ruiz, I.R., Cintas, H.P., Maldonado, L.J.H. (eds.) III Simposio de Inteligencia Computacional, SICO 2010, pp. 99–106. Ibergarceta Publicaciones, S.L., Madrid (2010)
3. Levine, M.D., Nobel, P.B., Youssef, Y.M.: A rule-based system for characterizing blood cell motion. In: Huang, T.S. (ed.) Image sequence processing and dynamic scene analysis. Springer, Heidelberg (1983)
4. Tostsos, J.K., Mylopoulos, J., Corvey, H.D., Zucker, S.W.: A framework for visual motion understanding. IEEE Trans. PAMI 2(6), 563–573 (1980)
5. Brémond, F., Medioni, G.: Scenario recognition in airborne video imagery. In: CVPR 1998 Whorkshop on Interpretation of Visual Motion, pp. 211–216. IEEE Computer Society, Los Alamitos (1998), ISBN 0-8186-8497-6
6. Medioni, I.C.a.G.: Detecting and tracking moving objetcs for video surveillance. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 1999), June 23-25, vol. 2, pp. 325–331 (1999) ISBN
7. Medioni, G., Cohen, I., Brémond, F., Hongeng, S., Nevatia, R.: Event detection and analysis from video streams. IEEE Trans. on Pattern Analysis and Machine Intelligence 23(8), 873–889 (2001)
8. Hongeng, S., Nevatia, R.: Large-scale event detection using semi-hidden markov models. In: IEEE Int. Conf. on Computer Vision (ICCV 2003), Nice, France, pp. 1455–1462. IEEE Computer Society, Los Alamitos (2003)
9. Hongeng, S., Nevatia, R., Brémond, F.: Video-based event recognition: activity representantion and probabilistic recognition methods. Computer Vision and Image Understanding (96), 129–162 (2004)

10. Natarajan, P., Nevatia, R.: Coupled hidden semi markov models for activity recognition. In: Workshop on Motion and Video Computing, pp. 10–17. IEEE Computer Society, Los Alamitos (2007)
11. Rivas-Casado, A., Martinez-Tomás, R.: Knowledge and event-based system for video-surveillance tasks. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5602, pp. 386–394. Springer, Heidelberg (2009)
12. Fernández-Caballero, A., Castillo, J.C., Rodríguez-Sánchez, J.M.: A proposal for local and global human activities identification. In: Perales, F.J., Fisher, R.B. (eds.) AMDO 2010. LNCS, vol. 6169, pp. 78–87. Springer, Heidelberg (2010)

Proposal for Extending New Linked Data Rules for the Semantic Web

Rafael Martínez Tomás and Luis Criado Fernández

Dpto. Inteligencia Artificial. Escuela Técnica Superior de Ingeniería Informática,
Universidad Nacional de Educación a Distancia,
Juan del Rosal 16, 28040 Madrid, Spain
rmtomas@dia.uned.es

Abstract. Semantic content annotations are not enough to construct the Semantic Web; these semantic data need to be linked. This is what the Linked Data fourth rule covers. Apparently, only the author of the content can do this task, but in this article we explore the possibilities of semantic technology and we study whether it is possible to create some types of semantic links more automatically, without intervention by the author. Furthermore, we study the problem of guaranteeing that the annotation in fact represents the content to which it refers, bearing in mind that the annotation should be as immediate as possible. As a result of these considerations, two new rules are proposed that make very clear the need to develop tools that automate the common-ontology based data link, thereby facilitating the update, security and consistency of the semantic information.

Keywords: Semantic Web, Semantic Views, RDFa, OWL, SPARQL, semantic annotation, Linked Data.

1 Introduction

In the last two years a critical mass of formally annotated information has been generated with semantics, based on the Linked Data concept [1]. The Semantic Web not only consists of publishing formally annotated data on the Web, but also of linking them with others, so that people and computer systems can explore the web of data and obtain related information from other initial data. It is in this context where the concept of "Linked Data" arises. In fact, it is a logical evolution of the foundational concept of the Web, the hyperlink, towards the formalisation and automation that the Semantic Web adds. It is the data link that gives the semantic web its value and also its power as a distributed computational knowledge system.

There are four rules that define this concept:

1. Using URIs (Uniform Resource Identifiers) as unique names for the resources.
2. Using the protocol HTTP to name and determine the location of the data identified with these URIs.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 531–539, 2011.

3. Offering information on the resources using RDF.
4. Including links to other URIs to locate more linked data.

The first three rules refer to the same elements that must be considered in the semantic annotation process. The fourth rule, which in fact gives the data link a name, prescribes using hyperlinks to external semantic information from the website whose content is being formally represented. This article analyses the current trend in Linked Data developments, which implies analysing certain aspects of the semantic annotation strategy. We raise several considerations (section 2) that start fundamentally from the problem of consistency and updating of semantic content annotation from the current authorship of the annotations and link. These considerations will later lead us to make some proposals for extending the Linked Data rules (section 3) that aim to help the aforementioned problems by improving the update frequency of the semantic data and introducing coherence between content and formally represented data. The evaluation of these ideas (section 4) is based on the results obtained with the sw2sws tool , which is a tool that populates ontologies that have been located as similar or approximate from html content pages. A description is also given of how generated semantic websites (rule 1, 2 and 3) can be used by a semantic search engine (in our case Vissem) to obtain the data link indirectly (rule 4). Finally, the article ends with some conclusions.

2 Considerations about the Current Four Rules

The Semantic Web has to interconnect the semantic contents, as the Linked Data fourth rule establishes. Yet in addition to this, several aspects that are key for the generalised acceptance of this concept and for (extended) migration towards a Semantic Web should be considered.

Consideration 1: link authorship
The simplest way of producing linked data consists of using a file, an URI that points to another. When an RDF file is written, for example[1], <http://example.criado.info/smith>, local identifiers can be used in it, so that we could refer to the identifiers #albert, #brian and #carol, which in N3[2] notation we will express as: <#albert> fam:child <#brian>, <#carol>.

And in RDF as:

```
<rdf:Description about="#albert"
<fam:child rdf:Resource="#brian">
<fam:child rdf:Resource="#carol">
</rdf:Description>
```

The current architecture of the WWW provides a global identifier "http://example.criado.info/smith#albert" for “Albert”, i.e. anyone can use this identifier to refer to “Albert” and thus provide more information. For example,

[1] http://www.w3.org/DesignIssues/LinkedData.html

[2] http://www.w3.org/DesignIssues/Notation3

in the document <http://example.criado.info/jones> someone could write: <#denise> fam:child <#edwin>, <smith#carol>.

or in RDF/XML:

```
<rdf:Description about="#denise"
<fam:child rdf:Resource="#edwin">
<fam:child rdf:Resource="http://example.criado.info/smith#
    carol">
</rdf:Description>
```

Just with this we have a basic Semantic Web in the sense that the semantic data are linked by the author of the contents. As a consideration it could be argued that this model is an exact reflection of the traditional web pages, where the authors of the contents decide the links.

Consideration 2*:* Embedded annotation

The current trend is to approach semantic annotation in an embedded way. Standards like HTML 5[3] , RDFa[4] and "XHTML+RDFa 1.1"[5] support this embedded procedure, and implementations like Linkator[2], the adoption of ontologies like GoodRelations[6] by Google[7] based on the use of "snippet" obviously confirm the firmness of this W3C proposal.

However, embedded annotation has to solve the problem of guaranteeing the coherence of the content that it represents. To explain this problem, the following must be considered: firstly, we have to be able to do a semantic annotation, i.e. formally represent a content, for example, of an HTML page in accordance with one or several ontologies related to that content. Secondly, the actor that uses or exploits this information now with semantics, for example, a search engine, must be sure that the annotations that it processes are coherent with the original content. If the annotation is embedded, then the HTML page incorporates semantics. What happens if something changes in the content affecting the embedded annotation? What happens if something changes in the content not affecting the embedded annotation? In both instances, if we calculate the HTML file digital signature with a hash function, it is verified that the signature has changed, so we can never be sure whether the "snippet" in RDFa representing its semantic content in fact still matches the HTML page. There is no way of guaranteeing coherence between content and annotation. In other words, because the annotation is inseparable from the HTML page the annotation update is uncertain.

As a consideration, embedded annotation has the disadvantage of not being able to guarantee coherence between the content and what has formally been expressed in RDFa. Consequently, developing the Semantic Web in such a weak model of inconsistency does not seem the most appropriate.

[3] http://www.w3.org/TR/html5/

[4] http://www.w3.org/TR/2010/WD-rdfa-core-20100803/

[5] http://www.w3.org/TR/2010/WD-xhtml-rdfa-20100803/

[6] http://www.heppnetz.de/projects/goodrelations/

[7] http://www.google.com/support/webmasters/bin/answer.py?answer=186036

Consideration 3: Frequency of generating or updating semantic data
Another aspect that must be considered is the update frequency of the annotation and data link. A Semantic Web that is always obsolete with the current Web content cannot be presented. For example, the penultimate version of DBpedia, when writing this article (January 2011), was DBpedia 3.5[8] that formalises Wikipedia data until March 2010 and has been active until the first fortnight of January 2011 (approximately a 10-month lag with Wikipedia). DBpedia version 3.6[9] was launched in the last fortnight of January 2011 and incorporates data up to November 2010. In other words, processing the data in the DBpedia project seems to need more than a month in the most optimistic of scenarios. Such a Semantic Web out of date with the current Web cannot compete with the current Web and is unlikely to be accepted by the general public.

Semantic data must be generated as soon as possible, ideally at the same time as the content. So, this leads us to ask who must generate the semantic data.

3 Proposal for New Linked Data Rules

We could refine the current four rules or add another two rules to avoid the problems detected in earlier considerations:

Rule 5. Semantic annotation must guarantee coherence between the content and pseudo-immediate update.

To guarantee coherence between annotation and content, as rule 5 establishes, a non-embedded annotation strategy must be used. This external annotation must relate the content source to the very annotation to guarantee coherence between formal representation and content.

A simple way of doing this is to have a mechanism to generate URIs based, for example, on hash functions derived from the HTML page where the formal annotations are to be done, which is the third approach that Martin Hepp analyses [4] in relation to GoodRelations ontology annotation. Thus, it is possible to obtain a unique URI for the very annotation (xml:base element), i.e. from the formal representation of the HTML page content.

Note that this procedure is no good when the annotation is embedded, since it is impossible to calculate a hash function to define an URI and then introduce a "snippet" into the web page without modifying the web page hash again. This occurs because a hash function obtains a unique identifier given an input, a file in this instance. When the input changes, although only a symbol changes, the hash result is an identifier completely different from the former. Therefore, if we obtain an URI from the hash of the HTML file content and then do an annotation in RDFa (as an example of embedded annotation) where we use this calculated URI and include all the code (snippet) in the HTML page, then, as soon as that snippet is included we have altered the HTML file content. Consequently, if we calculate the URI again from the hash of the HTML file content,

[8] `http://lists.w3.org/Archives/Public/semantic-web/2010Apr/0111.html`

[9] `http://lists.w3.org/Archives/Public/semantic-web/2011Jan/0105.html`

we will obtain an URI that is completely different from the URI that we had previously embedded. This implies that the URI, although unique, never allows us to determine whether the embedded annotation is coherent with the content that it represents.

The second aspect of this same rule requires the annotation, coherent with the content, to be as updated as possible. This problem implies developing procedures that are as automatic as possible, in keeping with our earlier works [3][6], and generally with automatic annotation.

Rule 6. Urge the shared use of ontologies, since this allows the non-explicit or dynamic data link, the ABOX-TBOX link.

In a Web without semantics, the links are established by the author of the content (rule 4). Yet in a Web with semantics there is also an approach where the data link is established dynamically when a SPARQL[10] query is made. Declaring instances of a same ontology by different authors of different webs without any semantic data links between the webs does not imply that these data cannot be linked afterwards. If these semantic data, which have been generated from the same ontology, are collected or grouped by a third person who wishes to exploit this information, then when it is all centralised in a SPARQL EndPoint[11], class instances can be obtained regardless of their origin. Thus, a SPARQL query would generate, under these conditions, a data link dynamically.

In an environment where data can be represented formally and where the data are classified into concepts (classes) that are defined by some characteristics (properties) and by the relations of all these elements (restrictions), the auto-link of some semantic data is possible in their exploitation, particularly the link between instances and classes (ABOX-TBOX), leaving the link between instances (ABOX-ABOX) to the author of the content (rule 4).

Rule 6 is complementary to rule 4 and could even be seen as a refined extension of rule 4. This rule facilitates the process, since just with rule 4, it is apparently difficult to automate the data link process, a problem that disappears when this proposal is improved.

To illustrate the meaning of the 6 rules together (including the proposed ones), we present the following example. Let us assume that on a web page mascotas (pets)[12] we annotate formally that bengala (bengal) is a gato (cat). We have an ontology to do this vertebrados (verteberates)[13], but in order to know that this annotation matches the content we apply MD5[14], which is a hash function, to fulfil rule 5: a509d1fdbeba807da648b83d45fd8903. Thus we could represent OWL formally as follows (line numbers have been added for the explanations):

[10] http://www.w3.org/TR/rdf-sparql-query/

[11] Un "EndPoint" indica una ubicación específica para acceder a un servicio Web mediante un protocolo y formato de datos específico. http://www.w3.org/TR/ws-desc-reqs/#normDefs

[12] http://www.mascotas.org/tag/el-gato-bengala

[13] http://www.criado.info/owl/vertebrados_es.owl

[14] http://tools.ietf.org/html/rfc1321

```
1  <rdf:RDF
2  xmlns:j.0="http://www.criado.info/owl/vertebrados_es.owl#"
3  xmlns:protege="http://protege.stanford.edu/plugins/owl/
       protege#"
4  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
5  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
6  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
7  xmlns:owl="http://www.w3.org/2002/07/owl#"
8  xmlns="http://www.mascotas.org/tag/el-gato-bengala#"
9  xml:base="http://example.criado.info/
       a509d1fdbeba807da648b83d45fd8903.owl">
10 <owl:Ontology rdf:about="">
11 <owl:imports rdf:resource="http://www.criado.info/owl/
       vertebrados_es.owl#"/>
12 </owl:Ontology> row 13: <j.0:gato rdf:ID="bengala"/> row
       14: <owl:AllDifferent>
13 <owl:distinctMembers rdf:parseType="Collection"> row 16: <
       gato rdf:about="#bengala"/>
14 </owl:distinctMembers>
15 </owl:AllDifferent>
16 </rdf:RDF>
```

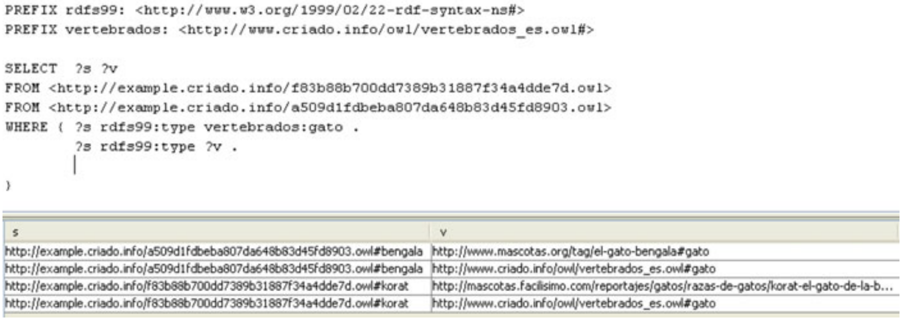
The resource defined bengala fulfils rule 1, because it has an URI (see row 9) which, of course, is unique. Furthermore, the coherence between content and annotation that rule 5 defines is also fulfilled in row 9, since it uses the MD5 algorithm derived from the HTML page of row 8. Rule 2 is fulfilled with row 8, because it uses the HTTP protocol to name and locate the data.

Continuing with the example, we need another similar file to show the ABOX-TBOX data link as:

```
1  <rdf:RDF
2  xmlns:j.0="http://www.criado.info/owl/vertebrados_es.owl
       #"
3  xmlns:protege="http://protege.stanford.edu/plugins/owl/
       protege#"
4  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
5  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
6  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
7  xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns="http://
       mascotas.facilisimo.com/reportajes/gatos/razas-de-
       gatos/korat-el-gato-de-la-buena-suerte_185682.html#"
8  xml:base="http://example.criado.info/
       f83b88b700dd7389b31887f34a4dde7d.owl">
9  <owl:Ontology rdf:about="">
10 <owl:imports rdf:resource="http://www.criado.info/owl/
       vertebrados_es.owl#"/>
11 </owl:Ontology>
12 <j.0:gato rdf:ID="korat"/>
13 <owl:AllDifferent>
14 <owl:distinctMembers rdf:parseType="Collection">
```

```
<gato rdf:about="#korat"/>
</owl:distinctMembers>
</owl:AllDifferent>
</rdf:RDF>
```

With both files, we could execute the following SPARQL query:



```
PREFIX rdfs99: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vertebrados: <http://www.criado.info/owl/vertebrados_es.owl#>

SELECT  ?s ?v
FROM <http://example.criado.info/f83b88b700dd7389b31887f34a4dde7d.owl>
FROM <http://example.criado.info/a509d1fdbeba807da648b83d45fd8903.owl>
WHERE { ?s rdfs99:type vertebrados:gato .
        ?s rdfs99:type ?v .
        
}
```

s	v
http://example.criado.info/a509d1fdbeba807da648b83d45fd8903.owl#bengala	http://www.mascotas.org/tag/el-gato-bengala#gato
http://example.criado.info/a509d1fdbeba807da648b83d45fd8903.owl#bengala	http://www.criado.info/owl/vertebrados_es.owl#gato
http://example.criado.info/f83b88b700dd7389b31887f34a4dde7d.owl#korat	http://mascotas.facilisimo.com/reportajes/gatos/razas-de-gatos/korat-el-gato-de-la-b...
http://example.criado.info/f83b88b700dd7389b31887f34a4dde7d.owl#korat	http://www.criado.info/owl/vertebrados_es.owl#gato

Obtaining the breeds of cats from the two different websites that have not established any explicit semantic data link, but that a third person (for example, us) has exploited, for example, with twinkle (a SPARQL query tool), and we have linked the ABOXs with the TBOXs. Rule 6 is thus fulfilled.

To generalise the former query, logically we cannot specify FROM of all the OWL files containing cats. "SPARQL EndPoints" must be used. Assuming that in some URL we had a SPARQL EndPoint storing semantic data on cats, fed with some LDSpider tool[5], we could make the following query:

```
PREFIX rdfs99: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vertebrados: <http://www.criado.info/owl/vertebrados_es
    .owl#>
SELECT ?s ?v
WHERE { ?s rdfs99:type vertebrados:gato . ?s rdfs99:type ?v }
```

Obtaining the list of all the cats that there are in different URLs of the Semantic Web that LDSpider would have been able to explore and incorporate into a database.

4 Conclusions

In this article we have considered Linked Data and Semantic Web concepts and their problems. A Semantic Web based on the currently accepted four rules provides a Semantic Web in deferred time, since this Semantic Web will always be out of date with the current Web, which is the source of information. We only have to check Linked Data platforms such as, for example, the DBpedia to discover that they are updated with subsequent versions and thus are not in real time[15]. This view of a Semantic Web behind the current Web is apparently enough of an obstacle to prevent full implementation.

[15] http://dbpedia.org/About

Obviously, the power of the current Web is primarily due to the fact that it is very simple to generate web pages, that the information is decentralized and that the Web is participatory. However, what makes the Web especially powerful and useful, is that the web pages are linked (the concept of hyperlink is inseparable from that of the web) and that publishing information and linking it with other information is very easy (publishing on the web as a task only requires an html or text editor, it is not even necessary to generate a 100% correct code). Work is currently being done on the idea that the information generated semantically (following the Linked Data rules) is linked by the same actor. This approach is the same as when we used HTML. Although our information can, of course, also be linked by a third person in both paradigms. These are explicit links. This characteristic of establishing explicit hyperlinks has been directly transferred to the Semantic Web environment and is the Linked Data fourth rule. Yet, in an environment where systems understand something of the content, is this as interesting? Who must establish the links? And what if the actor exploiting the semantic data could establish the links?

In this work, we have not only studied these problems, but we have also tried to provide new ideas that help direct the data link process to the most up-to-date Semantic Web possible with regard to the current Web. We have proposed introducing two new rules, which are fundamentally aimed at extending automatic annotation based on shared ontologies, so that the problems of consistency and updating of content annotation are solved. The proposals are accompanied with an application example of the 6 resulting rules.

Acknowledgements

The authors are grateful to the CiCYT for financial aid on project TIN2010-20845-C03-02.

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009), doi:10.4018/jswis.2009081901
2. Araujo, S., Houben, G.J., Schwabe, D.: Linkator: Enriching Web Pages by Automatically Adding Dereferenceable Semantic Annotations. In: Benatallah, B., et al. (eds.) ICWE 2010. LNCS, vol. 6189, pp. 355–369. Springer, Heidelberg (2010)
3. Criado, L.: PhD thesis. Semi-automatic Procedure for Transforming the Web into a Semantic Web. Universidad Nacional de Educa-ción a Distancia. Escuela Técnica Superior de Ingeniería Informática. Madrid, España (2009)
4. Gomez, M., Preece, A.D., Johnson, M.P., de Mel, G., Vasconcelos, W.W., Gibson, C., Bar-Noy, A., Borowiecki, K., La Porta, T., Pizzocaro, D., Rowaihy, H., Pearson, G., Pham, T.: An ontology-centric approach to sensor-mission assignment. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 347–363. Springer, Heidelberg (2008)

5. Isele, R., Umbrich, J., Bizer, C., Harth, A.: LDSpider: An open-source crawling framework for the Web of Linked Data. Poster at the International Semantic Web Conference (ISWC2010), Shanghai (November 2010), on line, http://www.wiwiss.fuberlin.de/en/institute/pwo/bizer/research/publications/IseleHarthUmbrichBizer-LDspider-Poster-ISWC2010.pdf
6. Fernández, L.C., Martínez-Tomás, R.: The Problem of Constructing General-Purpose Semantic Search Engines. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5601, pp. 366–374. Springer, Heidelberg (2009)

$AWARD^{prime}$: An Adaptive Web Based-Tool Prototype for Neurocognitive Individualized Assessment and Training

Raquel Salmerón[1], Serafín Crespo[1], Francisco López[2], María Teresa Daza[2], and Francisco Guil[1]

[1] Dept. de Lenguajes y Computación - Universidad de Almería
{rsalmeron,scp122,fguil}@ual.es
[2] Dept. Neurociencia y Ciencias de la Salud - Univeridad de Almería
{fll416,tdaza}@ual.es

Abstract. As the web is expanding and the underlying technologies are evolving, so does the way in which users interact with the web applications. It is becoming increasingly clear that web applications must be adapted to the specific characteristics of each user, resulting in what is known as Adaptive Web Applications. An obvious fact is that there is no a single user model. Actually, there are many factors, some quantifiable, which define how a user interacts with a particular web application, allowing the determination of the user profile and setting the associated default values of each adaptive variables. This paper presents a prototype tool whose core consists of an adaptive method based on similarity between (mined) event-based sequences. The application domain belongs to the area of cognitive neuroscience. In particular, the tool has been designed as a platform for the specification of adaptive cognitive tasks, created to assess and stimulate different aspects of cognition in patients with neurological damage.

1 Introduction

Learning and evolution are two fundamental forms of adaptation. The adaptive web emerges as a solution to satisfy, depending on the interaction with the web application, the particular necessities of each specific user [9]. An adaptive web application is defined as an application that automatically improve its organization and presentation by learning from users access patterns.

Usually, adaptive web applications mine the data buried in web server logs to produce more easily navigable web applications. We find in the literature several approaches for adapting the web structure depending of the particularities of each user. These approaches can be divided into two groups: explicit and implicit methods. The explicit methods determine user profiles based on an enumeration of predetermined rules. On the other hand, the implicit methods extract the user profiles from a set of discovered patterns from web server logs [8]. Following the same logic as the implicit methods, we present an adaptive web-based application

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 540–549, 2011.

for neurocognitive individualized assessment and training. However, the solution differs from the previous proposal in two ways. First, instead of mining the web server log, the user profile (that is represented by a sequence of events) is extracted from a complex database storing performance data from different patients. Second, in the data analysis for creating that profiles database we take into account the temporal dimension. Neurocognitive assessment is a dynamic domain, and therefore, if we want to obtain useful patterns, we must incorporate the temporal dimension in the data mining techniques. Also, we use a similarity method between sequences for obtaining the most representative user profile. In this case, the user profile is determined dynamically, based on comparison of the user performance pattern (modeled as an event-based sequence) with profiles database.

The main goal of the web tool presented in this work, called $AWARD^{prime}$, is to provide a framework in which the clinical neuropsychologist (expert of the domain) can design individualized programs for neurocognitive assessment and training in patients with cognitive impairment after brain injury. Brain injury may affect the cognitive functioning in many ways. Different cognitive abilities are located in different parts of the brain, so a head injury can damage some, but not necessarily all, skills such as speed of thought, memory, understanding, concentration, solving problems and using language. The patients with brain injury form a heterogeneous population, whose strengths and needs vary on an individual basis. As a result, the assessment and rehabilitation of cognitive skills in these patients can be a challenging prospect for the clinical neuropsychologist. Thus, when administering the programs for assessment and rehabilitation/training, the neuropsychologist should adjust the task parameters through trial and error techniques, where decisions are based on their personal and professional experience, assuming a big-time effort and dedication. The use of adaptive computing solutions for managing these programs, provides a great help to the expert in the process of making decisions about which are the values of those parameters most suitable for each individual patient.

In this paper we focus on a special type of neurocognitive tasks designed for the assessment and rehabilitation/training of working memory, one of the main components of human memory (WM) [3]. To accommodate this type of tasks to the characteristics of each patient we propose to establish a temporal adaptive parameter called *time delay*. The tool adapts the task by assigning dynamically the most appropriate value of the parameter time delay associated with the current user profile. In particular, the user behavior is modeled as a sequence of temporal events. The comparison and selection of the appropriate profile will be done by implementing a temporal similarity method between sequences of events. The profiles database, which represents the knowledge base of the application, will be composed by a set of sequences representing the prototypical behavior of each patient. These sequences will be obtained by combining data mining techniques and the knowledge extracted from the domain expert. Each sequence will be associated with the parameter time delay more appropriate to the specific characteristics of the profile that represents.

The remainder of the paper is organized as follows. Section 2 describes the application domain of the proposed tool. In Section 3 explains briefly adaptive web concept and details the most important aspects of our proposal adaptive architecture. Section 4 presents the $AWARD^{prime}$ tool and shows the basic functionally using some practical examples. And finally, in Section 5, we discuss the main conclusions and future work arising from this proposal.

2 Application Domain

This work is defined on the domain application known as Computer-assisted Neurocognitive Assessment and Rehabilitation in neurologic patients, belonging to the area of Cognitive Neuroscience. Much of work done in this field of application consist of design specific programs for assessing and rehabilitating cognitive functions that may be altered or lost after brain injury or a result of the onset of illness neurodegenerative.

One of the neurocognitive functions that are frequently affected in these patients are those related to working memory (*WM*). WM is a theoretical construct that refers to the mechanism or system underlying the maintenance and processing of task-relevant information during the performance of a cognitive task. WM allows several pieces of information to be held in mind simultaneously and interrelatedly. It is essential for complex cognitive processes, such as spoken and written language comprehension, mental arithmetic, reasoning, and problem solving [3]. WM is also a subcomponent of the overall memory system, allowing the temporary storage and manipulation of information necessary for complex tasks. In contrast to the overall system, WM is, however, limited in both its storage and processing capacity. WM and attention are closely related; WM is a prerequisite for the selection of relevant information to attend to and the filtering out of irrelevant information. Stroke-induced deficits in WM and attention are often severe and result in impairments to vocational performance and social functioning. The degree of WM impairment, for tasks that require WM, is crucial for predicting recovery from stroke [10] and predicting the probability of returning to work [2]. Because WM capacity is a fundamental cognitive ability upon which rehabilitation of other functions depend [3][5], deficits in WM are crucial to the treatment approach. All the above justifies the need to provide tasks to assessment the functioning of this type of memory, as well as specific tasks for its rehabilitation or training.

Nowadays is increasingly that the programs for neurocognitive assessment and rehabilitation/training are implemented using computer support [11]. In general terms, computerized neurocognitive tasks measure response time more precisely, require less administration time, include alternate forms, and are ideal for rapid screening/triage. However, as describes above, the target population is extremely heterogeneous, even with these standard computerized neuropsychological tasks, it is difficult to measure accurately cognitive functioning in patients with brain injury, specially because these patients often show a slow-down in speed of information processing ("cognitive slowing").

This "cognitive slowing" in each patient must be taken into careful consideration both during the administration of tasks and also during the process of analyzing the obtained results. The patients might show a deficient performance in a particular cognitive task because of this "cognitive slowing" even if the underlaying cognitive function remains intact. In this situation, and with the aim of controlling the effect of the speed of information processing, it is necessary to adapt the timed-parameter associated with the tasks, establishing the right values according with the patient profile.

In general, the WM tasks [12] are aimed to assess or to train the capacity of an individual to codify, maintain and manipulate information in WM. Briefly, WM tasks involved: (i) maintenance of multiple stimuli at the same time; and (ii) short delays during which the representation of stimuli should be held in WM. Figure 1 shows the general structure of a trial of a computerized WM task. Each square represent a computer screen.

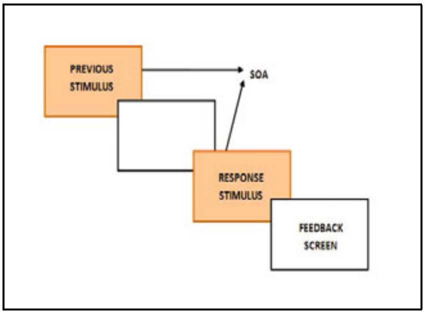


Fig. 1. Screenshot of configuration trial

Each trial of the task starts with an initial screen that contains a previous stimuli set. Normally the duration of this screen is proportional to the number of stimuli. After that, a screen remains blank during a short delay, denominated SOA. Then, appears another screen with the response stimuli set. This screen remains until the patient makes a response. At last a feedback screen is shown to the patient before continue with the next trial.

In this way, in each task, the adjustment of temporal parameters related to the duration of the events of a trial is a fundamental problem in such scenarios. The settings of these events can be summarized in the adjustment of two parameters. On the one hand the time that must be present the previous stimuli set and on the other hand, the time between the concealment of the previous stimuli set and the appearance of the response stimuli set (delay during which the representation of previous stimuli should be held in WM). Both time values are difficult to adjust without an analysis thorough of each task and historical tests.

3 Adaptive Web Tools

Traditionally, the concept of adaptive web tool has been associated more with the domain of web personalization and computer education, using as a data source the records server (logs), the set of metadata, databases, the structure of the own tool and the actual domain knowledge, often expressed as ontology. The rapid growth of Internet encompasses a growing range of activities, offering a multitude of features to different and varied user profiles, each one with specific needs that can not be satisfied on account of the static character of traditional web applications.

From the point of view, both architectural and algorithms, the revision made of different types of traditional approaches to development of the automatic customization of systems reveals limitations [7]. As noted by this study, the main disadvantage of these approaches is related to the concept of scalability due to the huge amount of information to be stored of each user. For this reason, the current review of the techniques used for the development of adaptive web systems has been focused on data mining techniques.

Data mining is an essential step in the process of knowledge discovery in databases (KDD) that consists of applying data analysis and discovery algorithms that produce a particular enumeration of structures over the data [4].

In our case, by combining data mining techniques, classic statistics techniques and knowledge gained directly from the domain expert, we have obtained a set of temporal patterns representing the overall behavior of a user. During the execution of a cognitive task in the tool, the detection of patient profile will take place in a dynamic way, looking at the profiles database of patients previously evaluated with a similar temporal evolution. In general, the concept of similarity is essential in the area of data mining, and in our case, will be the basis on which rests the adaptive nature of the proposed application. The idea of similarity is the fact that it is not sufficient to consider the equality or inequality between objects (data). We need to consider how similar are two objects. The significance of the similarity between two objects depends on the type of data being treated. In one set of data you can see various types of similarities, therefore, different measures of similarity may reflect different aspects of the data set, so depending on application of the similarity measure we have to analyze which to choose.

One possible approach is in which the similarity between objects is defined in terms of complementary concept of distance. A distance measure d, must be natural in some sense, and should describe the aspects of the data that we are interesting to see. So, determining the most appropriate user profile can be established by calculating the degree of similarity between sequences of events using the edit distance algorithm. This distance is defined as the number of editing operations (insertion, deletion and/or modification) required to transform a source sequence in a target sequence. To tailor the extent costs can be assigned to each type of editing operation for each type of event, establishing a hierarchy of types of events. To calculate the weighted edit distance between two sequences of events, we used a version of the algorithm presented by Pirjo and Mannila [6].

In the work presented in [1], we can see a formal description of our proposal architecture. To determine the values of the adaptive parameters best suited for a particular user, the tool compares the current pattern with each of the patterns stored in a profile database. Obtaining the user profile that is being evaluated is achieved through the implementation of a method of similarity between sequences of events.

The web dynamic tool AWARD^{prime}, has been developed based on the general architecture for an adaptive web of the work mentioned above. The adaptive nature of the Web application is characterized, precisely, both by the temporal data mining and the similarity technique used.

4 The *AWARD^{prime}* Adaptive Tool

In this section, we present *AWARD^{prime}*, the web application designed for adaptive management and execution of specific tasks of *WM*. These tasks consist of a variable number of trials formed by one or more sample stimuli (which may include images, sounds or videos), and one or more comparison stimuli (response), separated between a period of delay. Automatic adjustment of this time interval of delay according to the subject's actions is the main objective of the application, besides the management functions.

During the execution of a cognitive task, the application will communicate with a web service that is in charge of finding, in an instant of time, the ideal profile that fits the user. Then it informs the application with the defining characteristics associated to that profile (*delay interval*). Thus, based on data provided by the web service, the application will be able to vary its execution parameters adapting to the user who is performing the task.

The tool also implements a management system for such assessment and training tasks, so the therapists can create new tasks, design the trials that compose the task, manage the data of subjects, etc. In addition, the application maintains a record with the cognitive tasks already completed, to provide important information to therapists for comprehensive monitoring of each subject.

Below we will make a brief review of the technology used to develop the tool. In order to simplify the design of the graphical interface and make it as standardized as possible we have used a combination of JSP (JavaServer Pages) technology that can generate dynamic web content, along with the definition of own labels using the standard specification tags JSTL 2.0. To implement the business logic have been used Java Servlet objects for access and modification of the database. As database management system we utilized MySQL. For events of the application that require access to server, but not uncomfortable reload page, we used the jQuery library of JavaScript/AJAX. For the specification of the web service of sequence similarity have been used as various Java programming IDE tools combined in a GNU / Linux, as Java2SDK version 5.0 and associated tools. As web services engine for Java have been used Tomcat server with AXIS package.

Once the features and developing technologies have been described, let us see how *AWARD^{prime}* works. We are going to show in detail how to use the main functionalities of the tool from a user perspective. Once AWARD^{prime} is started, and after the authentication process, we access to the main screen of the application. Figure 2 shows this main page, where are accessible the features of tool, that are divided into two groups: administration section and execution tasks section. In the administration module, the therapists can store and modify data related to subjects and create new protocols and tasks. On the other hand is the execution module where subjects can perform the training tasks.

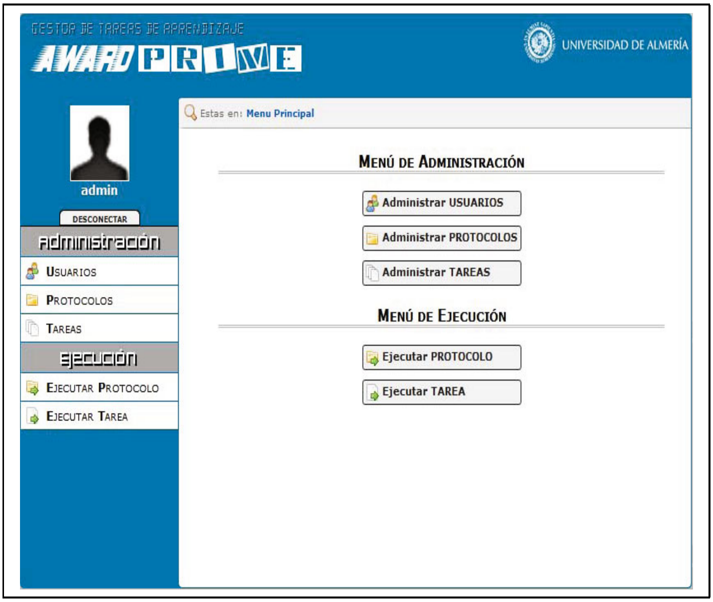


Fig. 2. Screenshot of main page

As an illustration of operation of the tool, we will work with the definition of two prototype tasks (*"Task of Nonsense Shapes"* and *"Task Board"*), which can be used to explore and/or training the visual-spatial sketchpad, a WM component involved in visual or spatial tasks, such as remembering shapes and colors or the location and speed of objects in space. To design a task, the therapist must define all the trials that compose it. Figures 3 and 4 illustrate the screens of the configuration form for each trial event (sample stimuli screen and choice stimuli screen) of the *"Task of Nonsense Shapes"*. Altogether, these two screens represent the configuration of a particular trial belonging to a task.

Briefly, we are going to describe the above cognitive tasks. The *"Task Nonsense Shapes"* is a delayed-match-to-sample task: the subject is shown a complex visual pattern (the sample) and then, after a brief delay (or retention interval in the range of 1 to 7 s -parameter adaptive time delay-), similar visual patterns (from 1 to 6 patterns). The subject must select the pattern which exactly matches the sample. Thus, in each trial there are two screens of events: the sample screen during 1 s, and the choice screen until response. After answering

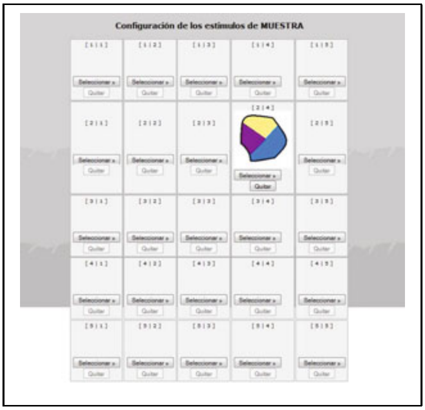


Fig. 3. Screenshot of trial configuration

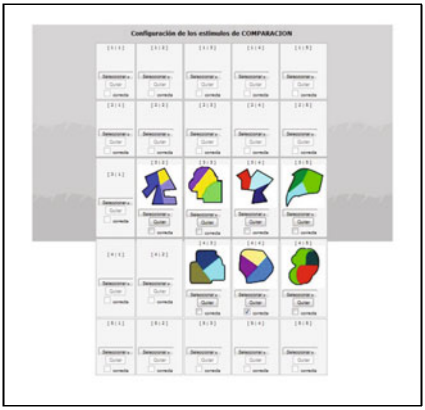


Fig. 4. Screenshot of response configuration

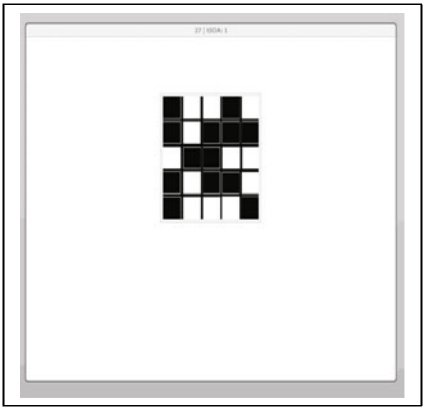


Fig. 5. Screenshot previous stimulus set

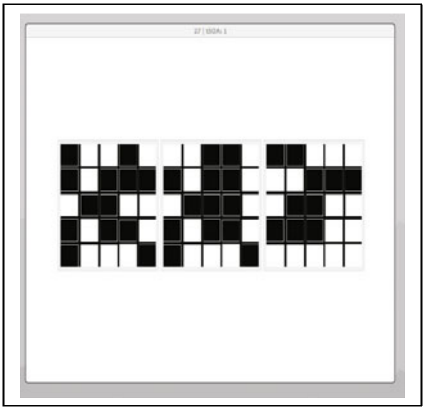
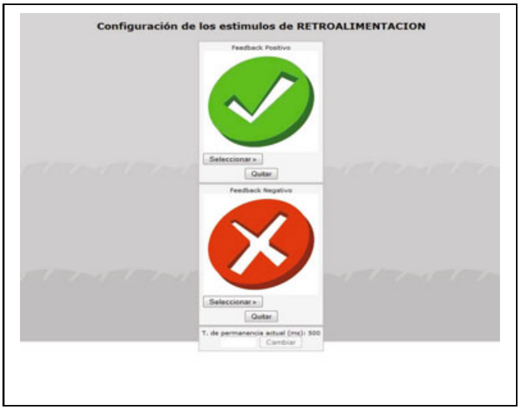


Fig. 6. Screenshot of response stimulus set

the user receives a visual feedback during 500 ms (see Figure 7). This task is composed of 48 trials, and the visual patterns consist of meaningless geometric shapes of different colors.

The *"Task Board"* is similar to above except that in the present task visual patterns consisting of 5x5 squares divided in 25 equal quadrants, some of which are colored (a "board"). In each trial there are two screens of events: the sample screen during 5 s (where the subject is shown a "board"), and the choice screen until response (where subject must select the "board" which exactly matches the sample "board").

At the beginning of the execution of a task, the subject should make a preset number of evaluation trials defined by the therapist. Then the system will ask to the similarity module for the most similar patient profile. At this point, the service accepts as input the sequence that models user behavior, and then


Configuración de los estimulos de RETROALIMENTACION

T. de permanencia actual (ms): 500

Fig. 7. Screenshot of feedback configuration

searches for the most similar sequence in the mined profiles database. Once the most similar profile is determined, the most suitable value for the *delay time* parameter is set, being the rest of trials adapted to the specific characteristics of the patient.

As an illustration of the execution of a task, Figures 5 and 6 present the events screens in a trial of the *"Task Board"* that are viewed by a user during the task execution.

Finally, Figure 7 displays the feedback screen associated to the trial just performed. There are two possibilities of configuration: positive feedback following successful execution of the trial, or otherwise negative feedback.

5 Conclusions and Future Works

In this paper we have presented $AWARD^{prime}$, an adaptive tool that allows distributed access and a simple end-user interaction, based on dynamic web forms with an efficient and intuitive interface. The usefulness of the tool is the creation of adaptive tasks for neurocognitive assessment and training, presenting two tasks designed to assess and train one of the main components of human memory, the working memory. As part of the decision-making module, we have described a web service that implements a similarity-based algorithm to determine user profiles, providing the basis for the adaptive functionality. The user profile is determined by comparing the sequence that models its behavior with a database of profiles mined from a historical dataset. As future work we propose two main lines: (1) use a more compact and informative kind of pattern, as possibilistic temporal patterns, and (2) extend the tool by incorporating new adaptive variables and increase the number of temporal milestones as a adjustment measure depending of the outcome.

Acknowledgments

This work is supported by the Grant P07-SEJ-03214 from Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía (Spain).

References

1. Antequera, M., Daza, M.T., Guil, F., Juárez, J.M., López-Crespo, G.: An architecture proposal for adaptive neuropsychological assessment. In: Proc. of IWINAC 2009. International Work-conference on the Interplay between Natural and Artificial Computation, pp. 426–436. Springer, Heidelberg (2009)
2. Brooks, D.N.: Measuring neuropsychological and functional recovery. In: Levin, H.S., Grafman, J., Eisenberg, H.M. (eds.) Neurobehavioral recovery from head injury, pp. 57–72. Oxford University Press, Oxford (1987)
3. Conway, A.R.: Workingmemory capacity and its relation to general intelligence. Trends Cogn. 7, 547–552 (2003)
4. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery: an overview. AIMagazine, pp. 37–54 (1996)
5. Malouin, F., Belleville, S., Richards, C., Desrosiers, J., Doyon, J.: Working memory and mental practice outcomes after stroke. Archives of Physical Medicine and Rehabilitation 85, 177–183 (2004)
6. Mannila, H., Ronkainen, P.: Similarity of event sequences. Proc. of the Int. Symposium on Temporal Representation and Reasoning (TIME 1997) (1997)
7. Mobasher, B.: Data mining for web personalization. In: The Adaptive Web, pp. 90–135 (2007)
8. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on web usage mining. Communications of the ACM 43, 142–151 (2000)
9. Etzioni, O., Perkowitz, M.: Towards adaptive web sites: Conceptual framework and case study. Artificial Intelligence 118(1-2), 245–275 (2000)
10. Robertson, I.H., Murre, J.: Rehabilitation of brain damage: Brain plasticity and principles of guided recovery. Psychological Bulletin 125, 544–575 (1999)
11. Sandford, J.A.: Cognitive training and computers: An innovative approach. In: Therapist's guide to learning and attention disorders, pp. 421–441. Academic Press, London (2003)
12. Sternberg, S.: Memory-scanning: Mental processes revealed by reactiontime experiments. American Scientific 57, 421–457 (1969)

Automated Mapping of Observation Archetypes to SNOMED CT Concepts

M. Meizoso[1], J.L. Allones[1], M. Taboada[1], D. Martinez[2], and S. Tellado[2]

[1] Department of Electronics and Computer Science, University of Santiago de Compostela, Spain
{maria.meizoso,joseluis.allones,maria.taboada}@usc.es

[2] Department of Applied Physics, University of Santiago de Compostela
{diego.martinez,serafin.tellado}@usc.es

Abstract. Integrating clinical data models, such as the European openEHR Archetypes and standard terminologies, such as SNOMED CT, is important to reduce medical errors and to get interoperability between health information systems. In this study, we propose an automated approach to mapping observation archetypes to SNOMED CT. Our approach applies a sequential combination of several basic matching methods, classically used in ontology matching. First, different lexical techniques identify similar strings between the observation archetypes and SNOMED CT. Second, a structure-based technique traverses two types of SNOMED CT relationships: *IS A* and *interprets*, searching for possible mappings not found with lexical techniques. The method was applied to the mapping of 20 observation archetypes. In total, 94% precision and 69% recall of SNOMED CT concepts was reached. Our method has revealed a degree of semantic similarity between some relationships in the tree representing observation archetypes and the relationships *IS A* and *interprets* in SNOMED CT.

Keywords: Knowledge Acquisition, Knowledge Representation, UMLS, SNOMED-CT.

1 Introduction

Nowadays, providing interoperability between health information systems is a critical issue for information sharing. On the one hand, standard codification of data is important to reduce medical errors. Terminologies, such as SNOMED CT [1] or LOINC [2], have been proposed as standard to code patient data. The aim is to avoid misinterpretation of patient data and the errors usual in traditional paper records. On the other hand, nowadays different organizations [3] are moving up the use of data entry forms to captured data in a structured way in Electronic Health Records (EHRs). Clinical data models, such as the European openEHR Archetypes [4], have been proposed for defining the structure of the information to be capture by these data entry forms and stored in EHR. Mapping terms of these clinical data models to a terminology system is a crucial step to provide interoperability between health information systems.

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 550–561, 2011.

However, the lack of automated methods simplifying the mapping process hampers interoperability. There is, among others, one problem making interoperability difficult: the excessive time required in manual processes, entailing exceptional and unfinished mappings. Therefore, automated methods to overcome this inconvenience are needed. There are some attempts in the literature, such as the MoST system [5], but more research is needed to provide useful tools.

In this paper, we propose an automated mapping method between an archetype content items and SNOMED CT concepts in order to offer an important aid to archetype construction support.

2 Materials

2.1 Archetypes

Archetypes provide clinicians with high-quality, re-usable clinical models of contents and processes together with formal interfaces to terminology. There are four types of archetypes to reflect common medical tasks for the clinicians interpretation: observations, evaluations, instructions and actions. Archetype are built using the Archetype Definition Language (ADL), which has been admitted as ISO draft, following the specifications of OpenEHR for an Electronic Health Record (EHR). An ADL file is comprised of several sections; the most important for this work are the showed in Fig. 1:

- **Definition section** containing the description of the archetype using constraints on instances of the types: HISTORY, EVENT, ITEM_LIST, ELEMENT, QUANTITY and CODED_TEXT.
- **Ontology section:** In this section, term codes are associated with text, description and bindings to other terminologies.

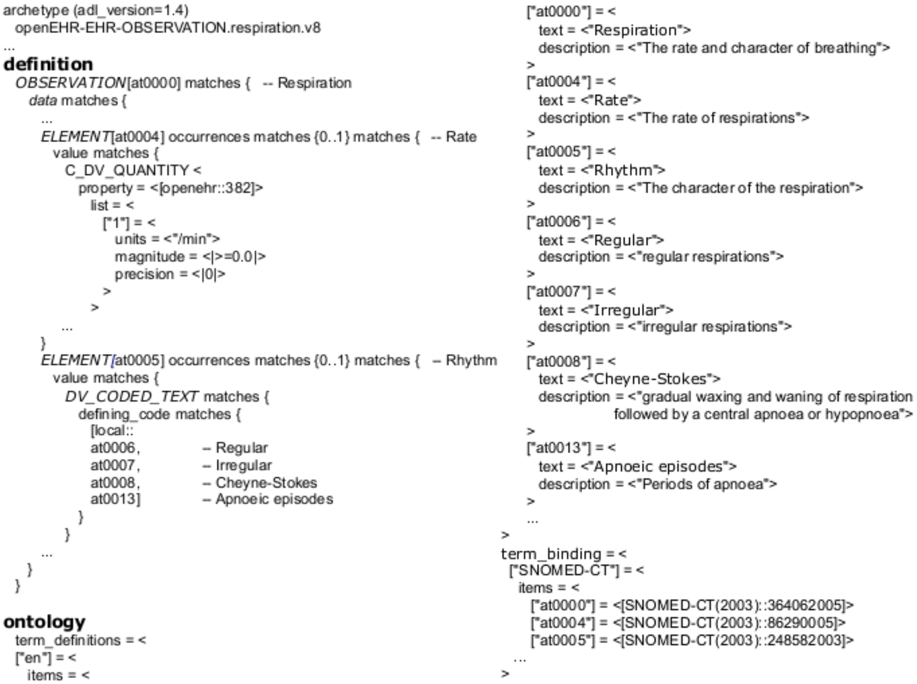
We work with the openEHR information model, and no others like CEN[6], HL7[7] or OMG HDTF[8], due to several reasons:

1. The openEHR Foundation is making an effort to integrate openEHR and SNOMED CT.
2. It is influenced by the other information models.
3. It fulfills the ISO, CEN and HL7 standards.

2.2 SNOMED CT

SNOMED CT consists of a set of concepts defined by descriptions and relationships with other concepts, such as:

- **The hierarchical relationship *IS A*:** All SNOMED CT concepts are descendants of a set of top level hierarchies.
- **Attribute relationships** between two concepts. There are different types of these relationships such as, the relationship *has interpretation* (grouped with the attribute *interprets*) which designates the judgement aspect being evaluated o interpreted for a concept (e.g. presence, absence, degree, . . .) An example of this type of relationship is shown in Table 1.



```
archetype (adl_version=1.4)
  openEHR-EHR-OBSERVATION.respiration.v8
...
definition
  OBSERVATION[at0000] matches {   -- Respiration
    data matches {
      ...
      ELEMENT[at0004] occurrences matches {0..1} matches {   -- Rate
        value matches {
          C_DV_QUANTITY <
            property = <[openehr::382]>
              list = <
                ["1"] = <
                  units = <"/min">
                  magnitude = <|>=0.0|>
                  precision = <|0|>
                >
              >
          ...
      }
      ELEMENT[at0005] occurrences matches {0..1} matches {   – Rhythm
        value matches {
          DV_CODED_TEXT matches {
            defining_code matches {
              [local::
              at0006,          – Regular
              at0007,          – Irregular
              at0008,          – Cheyne-Stokes
              at0013]          – Apnoeic episodes
            }
          }
      ...
    }
  }

ontology
  term_definitions = <
  ["en"] = <
    items = <
        ["at0000"] = <
          text = <"Respiration">
          description = <"The rate and character of breathing">
        >
        ["at0004"] = <
          text = <"Rate">
          description = <"The rate of respirations">
        >
        ["at0005"] = <
          text = <"Rhythm">
          description = <"The character of the respiration">
        >
        ["at0006"] = <
          text = <"Regular">
          description = <"regular respirations">
        >
        ["at0007"] = <
          text = <"Irregular">
          description = <"irregular respirations">
        >
        ["at0008"] = <
          text = <"Cheyne-Stokes">
          description = <"gradual waxing and waning of respiration
                      followed by a central apnoea or hypopnoea">
        >
        ["at0013"] = <
          text = <"Apnoeic episodes">
          description = <"Periods of apnoea">
        >
        ...
    >
    term_binding = <
     ["SNOMED-CT"] = <
       items = <
         ["at0000"] = <[SNOMED-CT(2003)::364062005]>
         ["at0004"] = <[SNOMED-CT(2003)::86290005]>
         ["at0005"] = <[SNOMED-CT(2003)::248582003]>
      ...
    >
```

Fig. 1. Respirations archetype

Table 1. Example of *interprets/has interpretation* relationship

Decreased skin turgor (finding)	INTERPRETS	Skin turgor (observable entity)
Decreased skin turgor (finding)	HAS INTERPRETATION	Decreased (qualifier value)

2.3 UMLS

The Unified Medical Language System (UMLS) [9] is a repository of biomedical ontologies and vocabularies developed by the US National Library of Medicine (NLM).Ontologies and Vocabularies integrated in the UMLS Metathesaurus include SNOMED CT, the Medical Subject Headings (MeSH), OMIM and the Digital Anatomist Symbolic Knowledge Base, etc. UMLS provides a mapping structure among these vocabularies.

We decided to use different tools provided by the NLM for mapping terms of the archetypes to SNOMED CT concepts for several reasons:

– UMLS integrates SNOMED CT and provides facilities for natural language processing and a set of Metathesaurus operations which are used to locate information efficiently in the Metathesaurus [10].

- The Metathesaurus includes many relations of synonymy (more than SNOMED CT), which increases the chances of successful searches. For example, the term *Cerebrovascular Stroke* is not covered in SNOMED CT. However, this is a synonym in UMLS.
- Archetypes designers often include words in texts of the nodes that hinder to find clinical concepts in terminologies. For example: *Description of Skin Color*. In these cases, the tool MetaMap [11] is a very useful tool as it is able to extract the clinical concept, *Color of Skin*, of the text.

3 Methods

We have developed and tested several mapping algorithms, classically used in ontology matching, to bind some terms included in the archetypes to the SNOMED CT. Our algorithms are focussed on mapping archetype terms to pre-coordinated SNOMED CT concepts. Several lexical and semantic techniques were applied to achieve the term mapping.

3.1 Preprocessing

In order to evaluate our approach, a set of test archetypes were selected from the repository created by NHS [12]. An ADL parser from the openEHR java reference implementation [13] was used to remove the details of archetype model syntax and preserve only the data hierarchy. In addition, only data nodes of the hierarchy with clinical meaning were selected for the mapping process input. Specifically, only the nodes *Element*, *Cluster*, *Point Event* and *Interval Event* were mapped to SNOMED CT concepts. For example, in the archetype in (Fig. 1) the nodes: *at0000*, *at0004*, *at0005*, *at0006*, *at0007*, *at0008*, *at0013* were selected.

The final result of parsing is a tree with different types of nodes: one root node, which contains the general observation recorded in the archetype (*Respiration* in Fig. 2), several *element* nodes which contain patient data related to the general observation (e.g. *Rate* and *Rhythm* in Fig. 2) and several *value* nodes which contain the possible values that the *elements* can take (e. g. *Regular*, *Irregular*, . . . in Fig. 2). The parser extracts the following information for each node: the identifier (*atxxxx*), the text of section term_definitions, the SNOMED CT concept identifier (if any) of section term_binding of the ontology and the type of node.

Once the trees are obtained by the parser for a set of archetypes, our method traverses these trees with the aim of mapping tree nodes. Firstly, the root node is mapped by applying a combination of matching methods, and then the rest of the nodes are mapped.

3.2 Mapping of the Root Node

In this section, different methods to mapping the root node of archetypes are detailed.

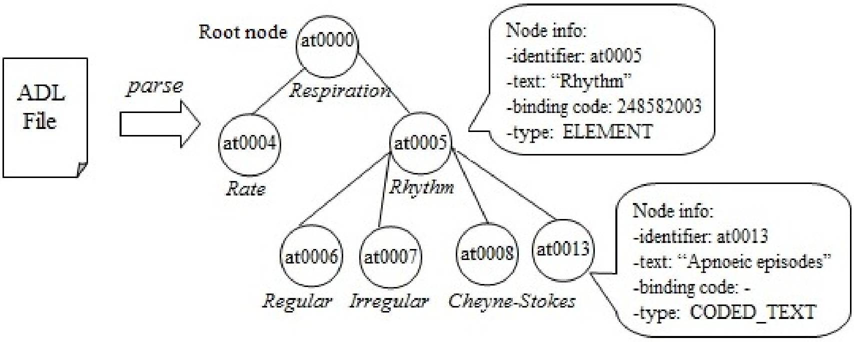


Fig. 2. Part of the tree resulting of parsing the ADL file shown in Fig 1

Mapping Through Metathesaurus and Metamap. Our approach sequentially uses different lexical matching techniques [14] until some mapping is reached. We initially applied an *Exact search* and, in case of no results we tried approximate techniques: *Truncate searches*, *Normalized word* searches and *MetaMap* (see Algorithm 1). Through this, we try to achieve a high recall without excessively penalizing the precision. Other types of searches, such as *Normalize String* or *Approximate Match*, have been discarded because we have checked that these worsen the results.

The next step involves getting the SNOMED CT concepts associated to the recovered UMLS concepts. Furthermore, the algorithm filters the SNOMED CT concepts belonging to the hierarchy *Observable Entity*, as in the work, the selected archetypes register observations. Finally, if two candidate SNOMED CT concepts share the same parent concept, this is added to the list of candidates. This optimization slightly increases the recall.

Algorithm 1. Archetype Root Node Mapping using Metathesaurus and MetaMap

```
FindSnomedConceptsForRootNode(Node rootNode)    begin
    $terms \longleftarrow \{rootNode.text, rootNode.text + “observable'', archetypeName\}$
    $candidateCUIs \longleftarrow$ FindByExactMatchingInMetathesaurus($terms$)
    if $|candidateCUIs| = 0$ then
        $candidateCUIs \longleftarrow$ FindByTruncateInMetathesaurus($terms$)
        if $|candidateCUIs| = 0$ then
            $candidateCUIs \longleftarrow$ FindByNormWordAndMetaMap($terms$)
    $candidateSnomedConcepts \longleftarrow$ getSnomedConcepts($candidateCUIs$)
    $candidateSnomedConcepts \longleftarrow$ addSharedFatherConcept($candidateCUIs$)
    return $candidateSnomedConcepts$
```

Direct Mapping to SNOMED CT. Our approach sequentially combines two basic types of matching methods classically used in ontology matching [15] in order to produce the alignment. First, we applied a "named-based method" that uses the lexical properties of archetype node text and SNOMED CT concept descriptions to find correspondences between them. We also used normalization techniques such as ignoring cases, skipping prepositions, avoiding initials in term names, and concatenating the word "observable" at the end of the node text. Once the candidate mapping set is recovered, a structure-based technique is applied between the archetype node tree and the hierarchy of candidate SNOMED CT concepts. First, the method computes lexical alignments between descendants of archetype root node and the narrower concepts of each candidate SNOMED CT concept. Second, the method computes the similarity value between candidate pairs (archetype node, SNOMED CT concept) by calculating the percentage of common descendant between the two components of the pair. Then, the method applies a conservative solution of the problem: it selects the mapping with the highest similarity value, rejecting the rest.

Algorithm 2. Archetype Root Node Mapping using SNOMED CT

```
MapRootNode(Node n)        begin
    $snomedCandidateConcepts \longleftarrow$ MapObservableConceptByName($n.text$)
    if $|snomedCandidateConcepts| = 0$ then
        $snomedCandidateConcepts \longleftarrow$
        MapObservableConceptByName($n.text$ + "observable")
    $rootDescendants \longleftarrow$ GetElementDescendantNodes($n$)
    $maxChildMapping \longleftarrow 0$
    for $candidate \in snomedCandidateConcepts$ do
        $candidateChildren \longleftarrow$ GetChildrenOfConcept($candidate$)
        $mappedChildrenCount \longleftarrow 0$
        for $descendant \in rootDescendants$ do
            $generatedNames \longleftarrow$ CombineNames($descendant.text$, $n.text$)
            for $child \in candidateChildren$ do
                for $name \in generatedNames$ do
                    if Map($child, name$) then
                        $mappedChildrenCount \longleftarrow mappedChildrenCount + 1$
        if $mappedChildrenCount > maxMappedChildren$ then
            $winners \longleftarrow candidate$
            $maxMappedChildren \longleftarrow mappedChildrenCount$
        if $mappedChildrenCount = maxMappedChildren$ then
            $winners$.add($candidate$)
        return $winners$
```

3.3 Mapping of the Element Nodes

Direct Mapping to SNOMED CT. All archetypes element node that have not been mapped to some SNOMED CT concept in Algorithm 2 are mapped in this stage. For each SNOMED CT candidate retrieved using the previous lexical matching, the method (Algorithm 3) traverses the relation *interprets* searching possible candidates for the values of the original element node (Fig. 3). After Algorithm 3, the non-mapped element nodes prospect for SNOMED CT correspondences using the previous "named-based" technique.

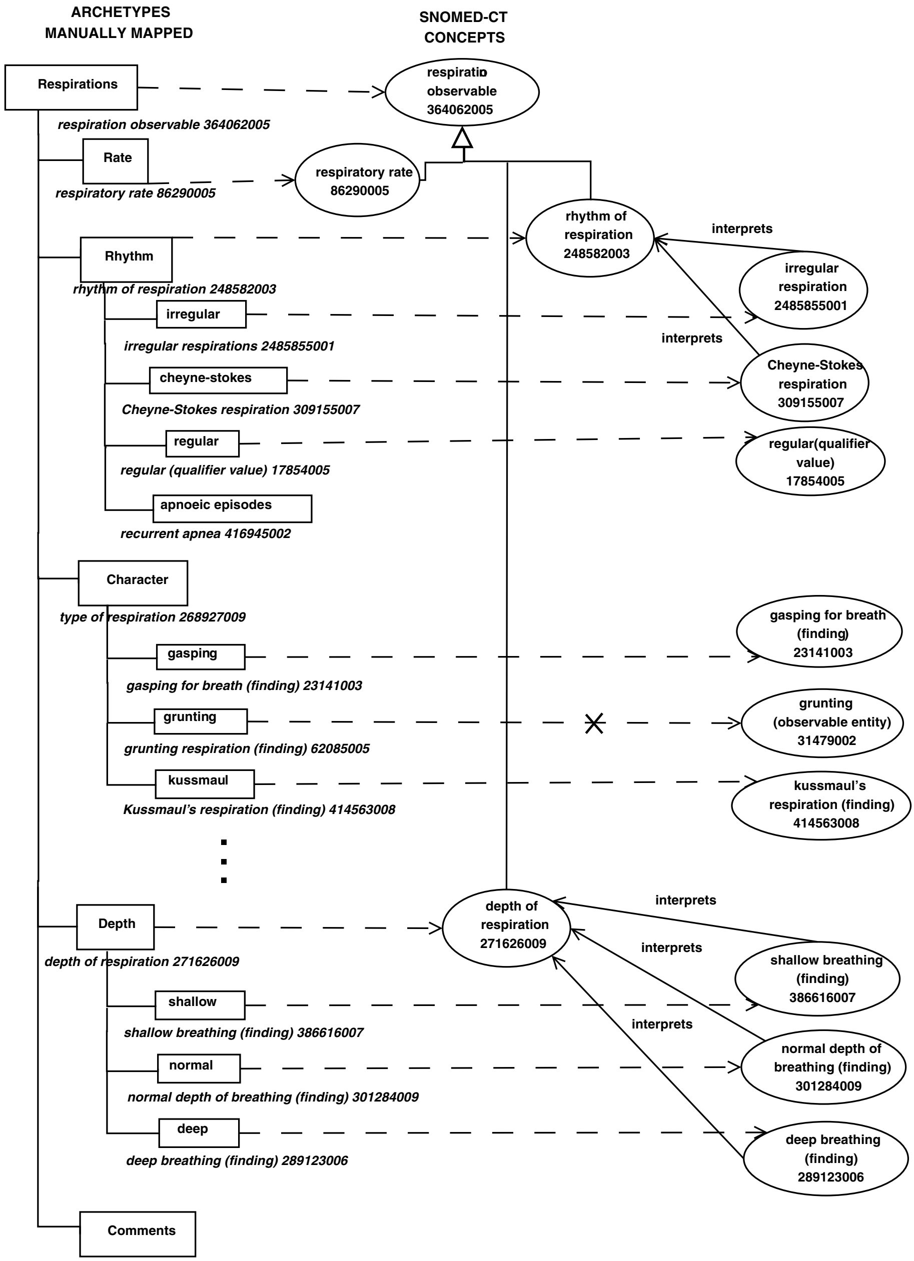

ARCHETYPES
MANUALLY MAPPED
SNOMED-CT
CONCEPTS
Respirations
respiratio observable 364062005
respiration observable 364062005
Rate
respiratory rate 86290005
respiratory rate 86290005
Rhythm
rhythm of respiration 248582003
rhythm of respiration 248582003
interprets
irregular
irregular respiration 2485855001
irregular respirations 2485855001
interprets
cheyne-stokes
Cheyne-Stokes respiration 309155007
Cheyne-Stokes respiration 309155007
regular
regular(qualifier value) 17854005
regular (qualifier value) 17854005
apnoeic episodes
recurrent apnea 416945002
Character
type of respiration 268927009
gasping
gasping for breath (finding) 23141003
gasping for breath (finding) 23141003
grunting
grunting (observable entity) 31479002
grunting respiration (finding) 62085005
kussmaul
kussmaul's respiration (finding) 414563008
Kussmaul's respiration (finding) 414563008
Depth
depth of respiration 271626009
depth of respiration 271626009
interprets
interprets
interprets
shallow
shallow breathing (finding) 386616007
shallow breathing (finding) 386616007
normal
normal depth of breathing (finding) 301284009
normal depth of breathing (finding) 301284009
deep
deep breathing (finding) 289123006
deep breathing (finding) 289123006
Comments

Fig. 3. Example of mapping element nodes and their values to SNOMED CT

Algorithm 3. Archetype Element Nodes Mapping using SNOMED CT

```
**MapElementNodes**(Node n)    **begin**
    $rootDescendants \longleftarrow$ GetNodeHierarchy($n$)
    **for** $descendant \in rootDescendants$ **do**
        **if** $|descendant.concepts| = 0$ **then**
            $snomedCandidateConcepts \longleftarrow$
            MapConceptByCombinedNames($descendant.text, n.text$)
            **for** $candidate \in snomedCandidateConcepts$ **do**
                **if** $|descendant.values| <> 0$ **then**
                    $values \longleftarrow descendant.values$
                    $mappedValuesCount \longleftarrow 0$
                    **for** $value \in values$ **do**
                        **if** GetInterpretsRelation($value, candidate$) $<> NULL$
                        **then**
                            $mappedValuesCount \longleftarrow mappedValuesCount + 1$
                    **if** $mappedValuesCount > maxMappedValues$ **then**
                        $winners \longleftarrow candidate$
                        $maxMappedValues \longleftarrow mappedValuesCount$
                    **if** $mappedValuesCount = maxMappedValues$ **then**
                        $winners$.add($candidate$)
            **return** $winners$
```

3.4 Mapping of Value Nodes

Direct Mapping to SNOMED CT. All archetype values that have not been mapped to some SNOMED CT in last step are mapped in this stage by applying a “named-based” method similar to the technique used in section 3.2 over all SNOMED CT.

3.5 Evaluation

The evaluation of mappings is the most critical part in terminology alignment research. The ideal technique to evaluate an alignment is to match it against a gold standard, that is, a reference alignment elaborated by a group of experts. In previous works [16], the bindings stored in some available archetype models [12] were used as gold standard. However, analyzing, in detail, the available archetype models, we have detected that they are exceptional and unfinished. So, we decided to create a gold standard by a quick manual lookup of SNOMED CT [17], [18] [19] for all selected archetype terms. Although creating our gold standard in this way is imperfect and not ideal, it is recommended in order to bring to light the strengths and limitations of an approach.

4 Results

Twenty observation archetype models, consisting of a total of four hundred and ninety-four nodes (terms), were selected from the NHS Connecting for Health

Table 2. Mapping results to SNOMED CT concepts

		Retrieved Concepts #	Retrieved Relevant Concepts #	Precision %	Recall %
ROOT NODE	Alg. 1: Exact Searches	13	11	85	55
	Alg. 1: Exact and Truncate Searches	55	17	31	85
	Algorithm 1: complete	87	20	23	100
	Algorithm 2	11	11	100	55
ELEMENT NODES	Algorithm 2	10	9	90	9
	Algorithm 3	55	50	91	51
VALUE NODES	Algorithm 3	6	6	100	4
	Algorithm 3 + mapping value nodes	127	121	95	83

project [3]. All of selected archetypes fulfill the following requirements: they are observable archetypes and their root nodes map to a SNOMED CT concept descendant of the top level concept *observable entity.* The reason of establishing this requirement is that we have found a correspondence between data sections nodes of this type of archetypes and the SNOMED CT concepts descendant of the top level *observable entity.*

Thirteen root nodes were lexically mapped to some SNOMED CT concept using the exact match Algorithm 1, although only eleven root nodes were correctly match. This represents 85% precision and 55% recall of the Algorithm 1 (Table 2). When approximate lexical techniques are applied (Algorithm 1: complete), 100% recall is reached (against 23% precision).

Two hundred and fifty-one element and value nodes were structurally mapped to some SNOMED CT concept. Nine element nodes were correctly mapped using the algorithm 2 with 90% precision and 9% recall. Applying algorithm 3, 50 element nodes were correctly mapped with 91% precision and 51% recall. These results confirm the need of applying a combination of named- and structure-based method in order to increase the recall. On the other hand, only six value nodes were correctly mapped using the algorithm 3 with 100% precision and 4% recall. When value nodes were directly mapped to SNOMED CT by applying a named-based method, 121 nodes were reached, with 95% precision and 83% recall.

For the root node mapping, we can see in table 2 that the Algorithm 1: complete (using UMLS) obtains a very high recall and a low precision. However, Algorithm 2 (using lexical matching) procures opposite results for precision and recall. For the other nodes, when the algorithm uses the SNOMED CT relationships (Algorithm 2 for element nodes and Algorithm 3 for value nodes), we obtain an important precision and low recall which is improved with lexical mapping techniques to SNOMED CT.

5 Discussion and Future Work

After creating the gold standard manually, we detected that 27.7% of nodes of the selected archetypes have no correspondence with SNOMED CT. There are two reasons for it:

- A 19.7% are nodes whose text are tagged with phrases like: *comment, additional information, any event, normal statement, row, new element* which are too general or their tag does not represent clinical information and, consequently, it have not correspondence with SNOMED concepts. We could skip mapping this kind of nodes because we extracted the list of avoidable phrases.
- The other 80.3% are nodes uncovered by SNOMED. Our method is focussed on mapping archetype nodes to pre-coordinated SNOMED-CT concepts. However, in some cases more than one SNOMED CT concept is needed to explain the archetype node. For example, the node *waist and hip circumference* will map to the SNOMED concepts *waist circumference (observable entity)* and *hip circumference (observable entity)*. So, post-coordination techniques are needed for mapping composite expressions of archetype nodes.

Other approaches [20], [16] and [21] use lexical and context methods for the mapping process. Our approach, in addition to the mentioned techniques, takes advantage of the SNOMED CT relationships *interprets* and *IS A*. On the one hand, our algorithm exploits the similarity between the relationships *element-values* of the archetypes nodes and the SNOMED CT relationship *interprets*. For example, in Fig. 3 the values *shallow*, *normal* and *deep* of the *element* node *depth* would never lexically match to the corresponding SNOMED CT concepts because of the word *breathing*. However, our algorithm can traverse the relation interprets of the SNOMED CT concept *depth of respiration*, and it applies partial lexical matching with the value nodes of element *Depth* in the archetype. On the other hand, the algorithm also profits from the correspondence between the relationships *root-elements* in archetype nodes and the SNOMED CT relationship *IS A*. For example (Fig. 3), after mapping the root node *respiration* to the SNOMED CT concept *respiration observable*, we use the *IS A* relationship for mapping the elements *depth*, *rate* and *rhythm* to the SNOMED CT concepts *depth of respiration*, *respiratory rate* and *rhythm of respiration*. In spite of obtaining a low recall with the relationship traversing application, the concepts retrieved offer a more appropriate semantics to the archetype. Therefore, this technique provides us a method of validation and disambiguation.

In order to increase the recall of our approach, in the future we plan to combine the lexical techniques used in the Algorithm 1 with structure-based techniques developed in the Algorithm 2 and 3. We will also explore more significant SNOMED CT relationships for validation and disambiguation.

6 Conclusions

In this study, we propose an automated approach to mapping terms identified in archetypes to SNOMED CT concepts. Our approach applies a combination of basic matching methods classically used in ontology matching[15]. First, several lexical techniques identify similar strings between the archetypes terms and SNOMED CT concepts. In parallel, a semantic technique processes mappings by extracting the relevant SNOMED CT subontology to the archetypes. The extraction of the subontology is carried out by covering the SNOMED CT relationships with the meaning probably more similar to the intended meaning of the archetype data. The methods were applied to the mapping of the twenty different Observation archetype models, with a total of four hundred and ninety-four selected archetype terms, to the complete SNOMED CT. In total, 94% precision and 69% recall was reached. Our method has revealed some degree of semantics in the structure of the information defined in the available archetypes with some SNOMED CT relationships, such as *interprets* and *IS A*. Future work will require to study more types of SNOMED CT relationships in order to guarantee our work.

Acknowledgements. This work has been funded by the Ministerio de Educación y Ciencia, through the national research project *Gestion de Terminologias Medicas para Arquetipos* TIN2009-14159-C05-05.

References

1. SNOMED-CT: Systematized Nomenclature of Medicine-Clinical Terms, http://www.ihtsdo.org/snomed-ct/
2. LOINC: Logical Observation Identifiers Names and Codes, http://loinc.org/
3. NHS: Connecting for Health project (2010), http://www.connectingforhealth.nhs.uk/
4. OpenEHR. archetypes., http://www.openehr.org/
5. Sundvall, E., Qamar, R., Nystrom, M., Forss, M., Petersson, H., Karlsson, D., Ahlfeldt, H., Rector, A.: Integrations of tools for binding archetypes to snomed ct. BMC Medical Informatics and Decision Making 8(Suppl. I), S7 (2008)
6. CEN. Europen Committee for Standarization, http://www.cen.eu/
7. HL7. Health level Seven International, http://www.hl7.org/
8. OMG. Omg Healthcare Domain Task Force, http://healthcare.omg.org/
9. UMLS. Unified Medical Language System,http://www.nlm.nih.gov/research/umls/
10. Metathesaurus, http://www.ncbi.nlm.nih.gov/books/NBK9684/pdf/ch02.pdf
11. MetaMap, http://mmtx.nlm.nih.gov/

12. NHS. NHS Connecting for Health Archetype Repositories, https://svn.connectingforhealth.nhs.uk/svn/public/nhscontentmodels/TRUNK/cm/archetypes/
13. OpenEHR. Parser, http://www.openehr.org/projects/java.html
14. Metathesaurus. Web service operations, https://uts.nlm.nih.gov//doc/devGuide/webservices.html#meta
15. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
16. Yu, S., Berry, D., Bisbal, J.: An investigation of semantic links to archetypes in an external clinical terminology through the construction of terminological "shadows". In: IADIS, International Association for Development of the Information Society, Germany (2010)
17. VTSL. SNOMED CT Core Browser, http://snomed.vetmed.vt.edu/sct/menu.cfm/
18. NCI. SNOMED CT Term Browser, http://nciterms.nci.nih.gov/ncitbrowser/pages/vocabulary.jsf?dictionary=SNOMED%20Clinical%20Terms&version=2010_01_31
19. UMLS. Terminology Services (beta). SNOMED CT Browser, https://uts.nlm.nih.gov/snomedctBrowser.html/
20. Qamar, R.: Semantic Mapping Of Clinical Model Data To Biomedical Terminologies To Facilitate Interoperability. PhD thesis, University of Manchester (2008)
21. Lezcano, L., Sanchez-Alonso, S., Sicilia, M.: Associating clinical archetypes trough umls metathesaurus term clusters. Journal of Medical Systems (2010)

Author Index