

António E. Ruano

Annamária R. Várkonyi-Kóczy (Eds.)

New Advances in Intelligent Signal Processing

António E. Ruano and Annamária R. Várkonyi-Kóczy (Eds.)

New Advances in Intelligent Signal Processing

Studies in Computational Intelligence, Volume 372

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 351. Ngoc Thanh Nguyen, Bogdan Trawiński, and Jason J. Jung (Eds.)
New Challenges for Intelligent Information and Database Systems, 2011
ISBN 978-3-642-19952-3

Vol. 352. Nik Bessis and Fatos Xhafa (Eds.)
Next Generation Data Technologies for Collective Computational Intelligence, 2011
ISBN 978-3-642-20343-5

Vol. 353. Igor Aizenberg
Complex-Valued Neural Networks with Multi-Valued Neurons, 2011
ISBN 978-3-642-20352-7

Vol. 354. Ljupco Kocarev and Shiguo Lian (Eds.)
Chaos-Based Cryptography, 2011
ISBN 978-3-642-20541-5

Vol. 355. Yan Meng and Yaochu Jin (Eds.)
Bio-Inspired Self-Organizing Robotic Systems, 2011
ISBN 978-3-642-20759-4

Vol. 356. Sławomir Koziel and Xin-She Yang (Eds.)
Computational Optimization, Methods and Algorithms, 2011
ISBN 978-3-642-20858-4

Vol. 357. Nadia Nedjah, Leandro Santos Coelho, Viviana Cocco Mariani, and Luiza de Macedo Mourelle (Eds.)
Innovative Computing Methods and their Applications to Engineering Problems, 2011
ISBN 978-3-642-20957-4

Vol. 358. Norbert Jankowski, Włodzisław Duch, and Krzysztof Grańbczewski (Eds.)
Meta-Learning in Computational Intelligence, 2011
ISBN 978-3-642-20979-6

Vol. 359. Xin-She Yang, and Sławomir Koziel (Eds.)
Computational Optimization and Applications in Engineering and Industry, 2011
ISBN 978-3-642-20985-7

Vol. 360. Mikhail Moshkov and Beata Zielosko
Combinatorial Machine Learning, 2011
ISBN 978-3-642-20994-9

Vol. 361. Vincenzo Pallotta, Alessandro Soro, and Eloisa Vargiu (Eds.)
Advances in Distributed Agent-Based Retrieval Tools, 2011
ISBN 978-3-642-21383-0

Vol. 362. Pascal Bouvry, Horacio González-Vélez, and Joanna Kolodziej (Eds.)
Intelligent Decision Systems in Large-Scale Distributed Environments, 2011
ISBN 978-3-642-21270-3

Vol. 363. Kishan G. Mehrotra, Chilukuri Mohan, Jae C. Oh, Pramod K. Varshney, and Moonis Ali (Eds.)
Developing Concepts in Applied Intelligence, 2011
ISBN 978-3-642-21331-1

Vol. 364. Roger Lee (Ed.)
Computer and Information Science, 2011
ISBN 978-3-642-21377-9

Vol. 365. Roger Lee (Ed.)
Computers, Networks, Systems, and Industrial Engineering 2011, 2011
ISBN 978-3-642-21374-8

Vol. 366. Mario Köppen, Gerald Schaefer, and Ajith Abraham (Eds.)
Intelligent Computational Optimization in Engineering, 2011
ISBN 978-3-642-21704-3

Vol. 367. Gabriel Luque and Enrique Alba
Parallel Genetic Algorithms, 2011
ISBN 978-3-642-22083-8

Vol. 368. Roger Lee (Ed.)
Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2011, 2011
ISBN 978-3-642-22287-0

Vol. 369. Dominik Ryżko, Piotr Gawrysiak, Henryk Rybinski, and Marzena Kryszkiewicz (Eds.)
Emerging Intelligent Technologies in Industry, 2011
ISBN 978-3-642-22731-8

Vol. 370. Alexander Mehler, Kai-Uwe Kühnberger, Henning Lobin, Harald Lüngen, Angelika Storrer, and Andreas Witt (Eds.)
Modeling, Learning, and Processing of Text Technological Data Structures, 2011
ISBN 978-3-642-22612-0

Vol. 371. Leonid Perlovsky, Ross Deming, and Roman Ilin (Eds.)
Emotional Cognitive Neural Algorithms with Engineering Applications, 2011
ISBN 978-3-642-22829-2

Vol. 372. António E. Ruano and Annamária R. Várkonyi-Kóczy (Eds.)
New Advances in Intelligent Signal Processing, 2011
ISBN 978-3-642-11738-1

António E. Ruano and Annamária R. Várkonyi-Kóczy
(Eds.)

New Advances in Intelligent Signal Processing

Editors

Prof. António E. Ruano
Universidade do Algarve
CSI / Centre for Intelligent Systems
Campus de Gambelas
8000 Faro
Portugal
E-mail: aruano@ualg.pt

Prof. Annamária R. Várkonyi-Kóczy
Óbuda University
Dept. of Mechatronics and Vehicle Engineering
Népszínház u. 8
1081 Budapest
Hungary
E-mail: varkonyi-koczy@uni-obuda.hu

ISBN 978-3-642-11738-1

e-ISBN 978-3-642-11739-8

DOI 10.1007/978-3-642-11739-8

Studies in Computational Intelligence

ISSN 1860-949X

Library of Congress Control Number: 2011933574

© 2011 Springer-Verlag Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

The *IEEE International Symposium on Intelligent Signal Processing (WISP)* event series celebrated its 10 years anniversary in 2009. The current volume “*New Advances in Intelligent Signal Processing*” contains extended works based on a careful selection of papers presented originally at the jubilee conference.

The importance of linking the scientific communities working in the fields of intelligent systems and signal processing was recognized in the late nineties by the IEEE Instrumentation and Measurement Society. Such facts that complex industrial and engineering systems, especially within the framework of large-scale embedded and real-time systems, confront researchers and engineers with completely new challenges, helped in setting up this new direction of science and research. Furthermore, it turned out that information processing and measurement is much more than originally meant: measurement and signal processing systems are involved in almost all kinds of activities in those fields, where control problems, system identification problems, industrial technologies, etc., are to be solved, i.e., when signals, parameters, or attributes must be measured, monitored, approximated, or estimated.

In a large number of cases, traditional information processing tools and equipment failed to handle the problems. Not only the handling of the previously unseen spatial and temporal complexity became questionable, but also problems had to be addressed such as the interaction and communication of subsystems based on entirely different modeling and information expression methods, the handling of abrupt changes within the environment and/or the processing system, the possible temporal shortage of computational power, and/or loss of some data, the uncertainty and ambiguity of the information, data, and perceptions, and last but not least, the new concepts of optimality and effectiveness.

The solution meant the introduction of new ideas for specifying, designing, implementing, and operating sophisticated signal processing systems. Computational intelligence, i.e., artificial intelligence, soft computing, anytime, and machine intelligence methods became serious candidates for handling many of the theoretical and practical problems, providing a better description, and, in many cases, proved to be the best if not the only alternatives for emphasizing significant aspects of system behavior.

Up until recently, however, these new techniques have not been widely used in the field of signal processing because some of the critical questions related to the design and verification have not been answered properly and because the uncertainty has been maintained in a quite different way as it is in the classical metrology.

These initiated the IEEE IM Society, the IEEE Hungary Section, and the European Association for Signal Processing to call to life a new event, hoping that it will become a series of regular meetings attracting more and more scientists and engineers in these hot topics. As result, the biannual symposium was launched in 1999.

Since that, five symposia have served as forum and catalyst of new theoretical and practical achievements in both the intelligent systems and signal processing communities. The continuous interest in WISP events has proved the soundness of the initiative, i.e., to link and counteract the scientific communities working in the fields of intelligent systems and signal processing.

The jubilee sixth IEEE International Symposium on Intelligent Signal Processing (WISP'2009), held in Budapest Hungary, August 26–28, 2009, contained 58 accepted papers out of the 81 submitted, from which 11 have been selected to incorporated in this volume. Present book does not intent to be an overall survey on the fields of interest of the area, but tries to find topics which represent new, hot, and challenging problems.

The book begins with papers investigating selected problems of *Modeling, Identification, and Clustering*. Chapter 1 presents fuzzy random variables as integral components of regression models. Chapter 2 introduces evolutionary multi-objective neural network models evolving optimized model structures that meet pre-specified design criteria in acceptable computing time. Chapter 3 deals with the structural learning model of neural networks within a Boltzmann machine. Chapter 4 proposes robust DNA-based clustering techniques and finally, in Chapter 5 the advances of combining multi-criteria analysis of signals and pattern recognition using machine learning principles are discussed.

In the second part of the volume *Image Processing* is treated. Chapter 6 deals with fuzzy relation based image enhancement, addressing also detail sharpening and noise cancellation. Chapter 7 describes an image contrast control technique based on the application of Łukasiewicz algebra operators. In Chapter 8, low complexity situational models of image quality improvement are presented. Chapter 9 proposes a flexible representation to map images to quantum computers. In Chapter 10 object recognition in images is addressed and weakly supervised classification models are proposed. The last chapter of the book, Chapter 11, presents an image processing application for elderly care, performing real-time 3D tracking based on a new evolutive multi-modal algorithm.

We would like to express out appreciation to all to the authors of this volume. Without their contributions we will not have this book in our hands. We are also grateful to the reviewers for offering their time in reviewing the papers. Their effort made possible to match the prestige of the books published by Springer Verlag.

A particular acknowledgment goes to Prof. Janusz Kacprzyk (Editor-in-Chief, Springer Studies in Computational Intelligence Series), Dr. Thomas Ditzinger (Senior Editor, Engineering/Applied Sciences Springer-Verlag), Ms. Heather King (Editorial Assistance, Springer Verlag, Heidelberg), and Ms. Teréz Anna Várkonyi (Editorial Assistance, Óbuda University, Budapest) for the editorial assistance and excellent collaboration during the development of this volume.

We hope that the reader will share our excitement and find the volume “*New Advances in Intelligent Signal Processing*” both inspiring and useful.

March 2011

Annamária R. Várkonyi-Kóczy
António E. Ruano
Budapest, Hungary and Faro, Portugal

Contents

1	Formulation of Fuzzy Random Regression Model	1
	Junzo Watada, Shuming Wang, Witold Pedrycz	
1	Introduction	2
2	Fuzzy Random Variables	4
3	Fuzzy Random Regression Model	7
4	The Solution to the FRRM	11
	4.1 Vertices Method	11
	4.2 Heuristic Method	13
5	An Example	15
6	Concluding Remarks	17
	References	18
2	Evolutionary Multiobjective Neural Network Models	
	Identification: Evolving Task-Optimised Models	21
	Pedro M. Ferreira, António E. Ruano	
1	Introduction	21
2	Methodology	23
	2.1 Problem Definition	24
	2.2 Multiobjective Evolutionary Algorithms	25
	2.3 Model Design Cycle	29
	2.4 ANN Parameter Training	30
3	Example Model Identification Problems	35
	3.1 Electricity Consumption Prediction	35
	3.2 Cloudiness Estimation	43
4	Concluding Remarks	50
	References	50
3	Structural Learning Model of the Neural Network and Its	
	Application to LEDs Signal Retrofit	55
	Junzo Watada, Shamshul Bahar Yaakob	
1	Introduction	56
2	Hopfield and Boltzmann Machine	57

3	Boltzmann Machine Approach to Mean-Variance Analysis	59
4	Double-Layered Boltzmann Machine Example	62
5	Overview on LEDs Signal Retrofit	64
6	LEDs Signal Retrofit and Mean-Variance Problem	65
7	Numerical Example of LEDs Signal Retrofit	67
8	Conclusions	72
	References	73
4	Robustness of DNA-Based Clustering	75
	Rohani Abu Bakar, Chu Yu-Yi, Junzo Watada	
1	DNA Computing Methods for Solving Clustering Problems	75
2	Background Study of Clustering Problems	76
3	Robustness of DNA-Based Clustering Algorithms	78
4	Proximity Distance Approach	80
5	Robustness in Clustering	82
	5.1 Dataset and Parameter Implementation	83
	5.2 Robustness Evaluation of DNA-Based Clustering	83
6	Results and Discussion	85
7	Conclusions	90
	References	91
5	Advances in Automated Neonatal Seizure Detection	93
	Eoin M. Thomas, Andrey Temko, Gordon Lightbody, William P. Marnane, Geraldine B. Boylan	
1	Introduction	93
2	Data and Experiment Setup	96
3	Probabilistic Classification Framework	97
	3.1 Preprocessing and Feature Extraction	98
	3.2 Classification	99
	3.3 Postprocessing	102
	3.4 Metrics	103
4	Results	104
	4.1 Results over All Patients	104
	4.2 Results for Individual Patients	107
5	Discussion	109
6	Conclusion	112
	References	112
6	Design of Fuzzy Relation-Based Image Sharpeners	115
	Fabrizio Russo	
1	Introduction	115
2	Linear Unsharp Masking: A Brief Review	116
3	Nonlinear Unsharp Masking Based on Fuzzy Relations	118
4	Effects of Parameter Settings	120
5	Noise Prefiltering Using Fuzzy Relations	122

6	A Complete Fuzzy Relation-Based Image Enhancement System	127
7	Conclusion	130
	References	130
7	Application of Fuzzy Logic and Lukasiewicz Operators for Image Contrast Control	133
	Angel Barriga, Nashaat Mohamed Hussein Hassan	
1	Introduction	133
2	Contrast Control Techniques	135
3	Soft Computing Techniques	138
	3.1 Minimization of Image Fuzziness	138
	3.2 Direct Method	139
	3.3 Fuzzy Histogram Hiperbolation	140
	3.4 Sharpening and Noise Reduction	140
4	Hardware Realizations	141
5	Contrast Control by Means of Lukasiewicz Operators	142
6	Control Parameters Based on Fuzzy Logic	144
7	Contrast Controller Architecture	149
8	Conclusions	152
	References	153
8	Low Complexity Situational Models in Image Quality Improvement	155
	Annamária R. Várkonyi-Kóczy	
1	Introduction	155
2	Corner Detection	156
	2.1 Overview	157
	2.2 Detection of Corner Points	159
	2.3 Experimental Results	162
3	“Useful” Information Extraction	163
	3.1 Overview	166
	3.2 Surface Smoothing	169
	3.3 Edge Detection	170
	3.4 Edge Separation	171
	3.5 Illustrative Example	172
4	A Possible Application: 3D Reconstruction of Scenes	175
5	Conclusions	176
	References	176
9	A Flexible Representation and Invertible Transformations for Images on Quantum Computers	179
	Phuc Q. Le, Abdullahi M. Iiyasu, Fangyan Dong, Kaoru Hirota	
1	Introduction	179
2	Flexible Representation of Quantum Images and Its Polynomial Preparation	181

3	Quantum Image Compression Based on Minimization of Boolean Expressions	186
4	Image Processing Operators on Quantum Images Based on Invertible Transformations	191
5	Experiments on Quantum Images	194
5.1	Storage and Retrieval of Quantum Images	194
5.2	Analysis of Quantum Image Compression Ratios	196
5.3	Simple Detection of a Line in a Quantum Image Based on Quantum Fourier Transform	198
6	Conclusion	199
	References	202
10	Weakly Supervised Learning: Application to Fish School Recognition	203
	Riwal Lefort, Ronan Fablet, Jean-Marc Boucher	
1	Introduction	203
2	Notations and General Framework	205
3	Generative Model	206
4	Discriminative Model	209
4.1	Linear Model	209
4.2	Non Linear Model	210
5	Soft Decision Trees and Soft Random Forests	211
5.1	Soft Decision Trees	211
5.2	Soft Random Forest	213
6	Classifier Combination	214
7	Application to Fisheries Acoustics	215
7.1	Simulation Method	215
7.2	The Fish School Dataset	216
7.3	Results	217
8	Conclusion	221
	References	221
11	Intelligent Spaces as Assistive Environments: Visual Fall Detection Using an Evolutive Algorithm	225
	José María Cañas, Sara Marugán, Marta Marrón, Juan C. García	
1	Introduction	225
2	Global System Description	228
3	Multimodal Evolutive Algorithm for Vision Based 3D Tracking	229
3.1	Explorers and Races	231
3.2	Fitness Function Observation Model	233
3.3	Determine 3D Positions	236

4	Experiments	237
4.1	Typical Execution	239
4.2	Time Performance	241
4.3	System Accuracy	242
5	Conclusions	245
	References	246
Author Index		249
Subject Index		251

Formulation of Fuzzy Random Regression Model

Junzo Watada*, Shuming Wang, and Witold Pedrycz

Abstract. In real-world regression analysis, statistical data may be linguistically imprecise or vague. Given the co-existence of stochastic and fuzzy uncertainty, real data cannot be characterized by using only the formalism of random variables.

To address regression problems in presence of such hybrid uncertain data, fuzzy random variables are introduced in this study, and serve as an integral component of regression models. A new class of fuzzy regression models based on fuzzy random data is built, and is called the fuzzy random regression model (FRRM). First, a general fuzzy regression model for fuzzy random data is introduced. Then, using expectations and variances of fuzzy random variables, σ -confidence intervals are constructed for fuzzy random input-output data. The FRRM is established based on the σ -confidence intervals. The proposed regression model gives rise to a non-linear programming problem which consists of fuzzy numbers or interval numbers. Since sign-changes in the fuzzy coefficients modify the entire programming structure of the solution process, the inherent dynamic non-linearity of this optimization makes it hard to exploit the techniques of linear programming or classical non-linear programming. Therefore, we resort to some heuristics. Finally, an illustrative example is provided.

Keywords: Fuzzy random regression model, Fuzzy regression model, fuzzy random variable, expected value, variance, confidence interval.

Junzo Watada · Shuming

Wang Graduate School of Information, Production and Systems,
Waseda University, 2-7 Hibikino, Wakamatsu,
Kitakyushu 808-0135, Fukuoka, Japan
e-mail: junzow@osb.att.ne.jp, smwangips@gmail.com

Witold Pedrycz

Department of Electrical & Computer Engineering, University of Alberta,
Edmonton, T6G 2G7, Canada

Witold Pedrycz

Systems Research Institute, Polish Academy of Sciences Warsaw, Poland
e-mail: pedrycz@ece.ualberta.ca

* Corresponding author. Mobile: +81-90-3464-4929. Fax: +81-93-692-5179.

1 Introduction

Classical regression model leads to effective statistical analysis of precise, numeric, statistical data. In the past two decades, to cope with imprecise data coming from fuzzy environments where human (expert) subjective estimates are used, various fuzzy regression models were introduced. The first fuzzy linear regression model, which treats fuzzy data instead of statistical data, was proposed by Tanaka *et al.* [29]. Tanaka *et al.* [31], Tanaka and Watada [32], Watada and Tanaka [38] presented possibilistic regression based on the concept of possibility measure. Chang [2] discussed a fuzzy least-squares regression, by using weighted fuzzy-arithmetic and the least-squares fitting criterion. Watada [40, 42] developed models of fuzzy time-series by exploiting the concept of intersection of fuzzy numbers. The limitation in these modelling pursuits is related with in dealing with fuzzy data with numeric inputs and fuzzy outputs.

To treat fuzzy input and fuzzy output data, Watada [41] proposed a heuristics-based fuzzy regression model which relies on heuristic methods to determine the products of fuzzy numbers. It was emphasized in [41] that the concepts of fuzzy statistics, fuzzy numbers and fuzzy arithmetic play a pivotal role in the design of fuzzy regression. More recently, some researchers have discussed fuzzy input-output data on the basis of nonlinear membership functions or nonlinear evaluation functions, which result in quadratic or non-linear programming problems. For example, Tanaka and Lee [33] discussed interval regression analysis based on quadratic programming instead of linear programming. Hao and Chiang [5] and Hong and Hwang [8] dealt with the nonlinear fuzzy model by exploiting support vector machines. Choi and Buckley [4] built fuzzy regression models based on the least absolute deviation estimators.

As an expansion of the models discussed in [31, 32, 38, 41], Watada and Mizunuma [43] and Watada and Toyoura [44] built switching fuzzy regression models to analyze mixed data obtained from various sources. Furthermore, linguistic regression models were discussed by Toyoura and Watada [34] and Watada and Pedrycz [45].

There have been many tests of real-world applications of fuzzy regression models, e.g., estimation of heat tolerance in plants [3], energy loss modeling [7], R&D project evaluation [9], peak load estimation of power distribution systems [20], modeling deregulated electricity markets [24], short-term load forecasting of power system [27], and reliability assessment [48]. In addition to fuzzy regression analysis, some other statistical approaches to fuzzy data can be found in Kandel [10], Nguyen and Wu [23], and Sun and Wu [28].

In contrast to the research noted above, this work is primarily concerned with regression models with hybrid uncertainty, where both fuzziness and randomness play a central role. Albeit randomness and fuzziness treated en bloc has been a controversial issue, the topic deserves attention bearing in mind that these two facets of uncertainty manifest quite often in real-world

problems. Zadeh [49] provided essential concepts for dealing with uncertainty under probabilistic and fuzzy environments.

The generalization of uncertainty theory was presented by Zadeh, where granularity and generalized constraints form the crux of the way in which uncertainty is being handled. Fuzzy random variables serve as basic tools for modeling optimization problems with such two-fold uncertainty. The concept of fuzzy random variable was introduced by Kwakernaak [12], who defined these variables as a measurable function linking a probability space with a collection of fuzzy numbers. Since then, various extensions of this concept have been developed, e.g., López-Díaz and Gil [17], Kruse and Meyer [11], Liu and Liu [15], Luhandjula [18], and Puri and Ralescu [25]. To deal with fuzzy random variables, a series of optimization models have been proposed, which help cope with uncertainty due to both randomness and fuzziness processed in an integrated fashion, e.g., fuzzy random multi-objective quadratic programming [1], fuzzy random goal programming [6], fuzzy random chance-constrained programming [13], two-stage fuzzy random programming [16], fuzzy random linear programming [37], fuzzy random reliability optimization models [35], and fuzzy random renewal processes [36].

Nevertheless, most of the existing studies on modelling fuzzy regression analysis have focused on data consisting of numeric values, interval-like numbers or fuzzy numbers without randomness into consideration. In practical situations, there exists a genuine need to cope with data that involves the factors of fuzziness and probability. For example, let us discuss experts' evaluation of products. Assume we have 100 samples of the same agricultural product. Suppose five inspectors (experts) evaluate the products on the basis of 10 attributes. Each expert grades each piece according to his experience and expertise. These gradings are given linguistically, e.g., "good", "very good", "bad" and "very bad", or about 5, about 6, and so on. When different inspectors give different grades, the grading is stochastic in its nature. That is, the differences among the five inspectors can be treated statistically, but each grade itself should be treated by considering the formalism of fuzzy sets. When we intend to build a multi-attribute model of the experts' evaluation, we have to consider this two-fold uncertainty, that is, uncertainty due to both fuzziness and randomness. Therefore, in the example considered here fuzzy random data should be employed to evaluate the products. Moreover, if we measure the change of the fuzzy random values using their confidence intervals, we can handle the multi-attribute problem by taking advantage of statistical analysis.

Motivated by the above reasoning, the objective of this paper is to design a fuzzy regression analysis technique, based on fuzzy random variables with confidence intervals, which will be referred to as fuzzy random regression analysis (FRRM). This study can be regarded as the generalization of our previous work [46], which focuses on a fuzzy regression model with an expected value approach to fuzzy random data. The confidence interval is defined by the expected value and variance of a fuzzy random variable. In

the realization of fuzzy random regression, it is difficult to calculate the product between a fuzzy coefficient and a confidence interval. We consider two approaches: a vertices method to describe the model, and a realistic heuristic method to solve optimization of the fuzzy random regression model.

The remainder of this paper is organized as follows. In Section 2, we cover some preliminaries of fuzzy random variables. Section III formulates the fuzzy random regression model. In Section IV, a solution to the problem is discussed. An explanatory example is provided to illustrate the proposed fuzzy random regression model in Section V. Finally, concluding remarks are given in Section 6.

2 Fuzzy Random Variables

Given some universe Γ , let Pos be a possibility measure defined on the power set $\mathcal{P}(\Gamma)$ of Γ . Let \mathfrak{R} be the set of real numbers. A function $Y : \Gamma \rightarrow \mathfrak{R}$ is said to be a fuzzy variable defined on Γ (see [19]). The possibility distribution μ_Y of Y is defined by $\mu_Y(t) = \text{Pos}\{Y = t\}$, $t \in \mathfrak{R}$, which is the possibility of event $\{Y = t\}$. For fuzzy variable Y with possibility distribution μ_Y , the possibility, necessity and credibility of event $\{Y \leq r\}$ are given, as follows

$$\begin{aligned} \text{Pos}\{Y \leq r\} &= \sup_{t \leq r} \mu_Y(t), \\ \text{Nec}\{Y \leq r\} &= 1 - \sup_{t > r} \mu_Y(t), \\ \text{Cr}\{Y \leq r\} &= \frac{1}{2} \left(1 + \sup_{t \leq r} \mu_Y(t) - \sup_{t > r} \mu_Y(t) \right). \end{aligned} \tag{1}$$

From (1), we note that the credibility measure is an average of the possibility and the necessity measure, i.e., $\text{Cr}\{\cdot\} = (\text{Pos}\{\cdot\} + \text{Nec}\{\cdot\})/2$, and it is a self-dual set function (see [14]), i.e., $\text{Cr}\{A\} = 1 - \text{Cr}\{A^c\}$ for any A in $\mathcal{P}(\Gamma)$. The motivation behind the introduction of the credibility measure is to develop a certain measure which is a sound aggregate of the two extreme cases such as the possibility (expressing a level of overlap and being highly optimistic in this sense) and necessity (articulating a degree of inclusion and being pessimistic in its nature). Based on credibility measure, the expected value of a fuzzy variable is presented as follows.

Definition 1 ([14]). *Let Y be a fuzzy variable. The expected value of Y is defined as*

$$E[Y] = \int_0^\infty \text{Cr}\{Y \geq r\} dr - \int_{-\infty}^0 \text{Cr}\{Y \leq r\} dr \tag{2}$$

provided that the two integrals are finite.

Example 1. Assume that $Y = (c, a^l, a^r)_T$ is a triangular fuzzy variable whose possibility distribution is

$$\mu_Y(x) = \begin{cases} \frac{x - a^l}{c - a^l}, & a^l \leq x \leq c \\ \frac{a^r - x}{a^r - c}, & c \leq x \leq a^r \\ 0, & \text{otherwise.} \end{cases}$$

Making use of (2), we determine the expected value of Y to be

$$E[Y] = \frac{a^l + 2c + a^r}{4}.$$

What follows is the definitions of fuzzy random variable and its expected value and variance operators. For more theoretical results on fuzzy random variables, one may refer to Gil *et al.*, Liu and Liu [15], and Wang and Watada.

Definition 2 ([15]). Suppose that (Ω, Σ, \Pr) is a probability space, \mathcal{F}_v is a collection of fuzzy variables defined on possibility space $(\Gamma, \mathcal{P}(\Gamma), \text{Pos})$. A fuzzy random variable is a mapping $X : \Omega \rightarrow \mathcal{F}_v$ such that for any Borel subset B of \mathfrak{R} , $\text{Pos}\{X(\omega) \in B\}$ is a measurable function of ω .

Let X be a fuzzy random variable on Ω . From the above definition, we know for each $\omega \in \Omega$, $X(\omega)$ is a fuzzy variable. Furthermore, a fuzzy random variable X is said to be positive if for almost every ω , fuzzy variable $X(\omega)$ is positive almost surely.

Example 2. Let V be a random variable defined on probability space (Ω, Σ, \Pr) . Define that for every $\omega \in \Omega$, $X(\omega) = (V(\omega)+2, V(\omega)-2, V(\omega)+6)_T$ which is a triangular fuzzy variable defined on some possibility space $(\Gamma, \mathcal{P}(\Gamma), \text{Pos})$. Then, X is a (triangular) fuzzy random variable.

For any fuzzy random variable X on Ω , for each $\omega \in \Omega$, the expected value of the fuzzy variable $X(\omega)$ is denoted by $E[X(\omega)]$, which has been proved to be a measurable function of ω (see [15, Theorem 2]), i.e., it is a random variable. Given this, the expected value of the fuzzy random variable X is defined as the mathematical expectation of the random variable $E[X(\omega)]$.

Definition 3 ([15]). Let X be a fuzzy random variable defined on a probability space (Ω, Σ, \Pr) . The expected value of X is defined as

$$E[\xi] = \int_{\Omega} \left[\int_0^{\infty} \text{Cr}\{\xi(\omega) \geq r\} dr - \int_{-\infty}^0 \text{Cr}\{\xi(\omega) \leq r\} dr \right] \Pr(d\omega). \quad (3)$$

Example 3. Consider the triangular fuzzy random variable X as defined in Example 2. Suppose that V is a discrete random variable, which takes values

$V_1 = 3$ with probability 0.2, and $V_2 = 6$ with probability 0.8. We calculate the expected value of X .

From the distribution of random variable V , we know that the fuzzy random variable X takes fuzzy variables $X(V_1) = (5, 1, 9)_T$ with probability 0.2, and $X(V_2) = (8, 4, 12)_T$ with probability 0.8. We need to compute the expected values of fuzzy variables $X(V_1)$ and $X(V_2)$, respectively. That is $E[X(V_1)] = (1 + 2 \times 5 + 9)/4 = 5$, and $E[X(V_2)] = (4 + 2 \times 8 + 12)/4 = 8$. Finally, by Definition 3, the expected value of X is $E[X] = 0.2 \cdot E[X(V_1)] + 0.8 \cdot E[X(V_2)] = 7.4$.

Definition 4 ([15]). Let X be a fuzzy random variable defined on a probability space (Ω, Σ, \Pr) with expected value e . The variance of X is defined as

$$\text{Var}[X] = E[(X - e)^2] \quad (4)$$

where $e = E[X]$ is given by Definition 3.

Example 4. Consider the triangular fuzzy random variable X defined in Example 3. Let us calculate the variance of X . From the distribution of random variable V , we know that the fuzzy random variable X takes fuzzy variables $X(V_1) = (5, 1, 9)_T$ with probability 0.2, and $X(V_2) = (8, 4, 12)_T$ with probability 0.8. From Example 3, $E(X) = 7.4$. Then $\text{Var}(X) = E[(X(V_1) - 7.4)^2] \cdot 0.2 + E[(X(V_2) - 7.4)^2] \cdot 0.8$.

To obtain $\text{Var}[X]$, we need to calculate $E[(X(V_1) - 7.4)^2]$ and $E[(X(V_2) - 7.4)^2]$, where $X(V_1) - 7.4 = (-2.4, -6.4, 1.6)_T$ and $X(V_2) - 7.4 = (0.6, -3.4, 4.6)_T$. Denoting $Y_1 = X(V_1) - 7.4 = (-2.4, -6.4, 1.6)_T$, first we will calculate $\mu_{Y_1^2}$ and $\text{Cr}\{Y_1^2 \geq r\}$. Since

$$\begin{aligned} \mu_{Y_1^2}(t) &= \text{Pos}\{Y_1^2 = t\} \\ &= \max\left\{\text{Pos}\{Y_1 = \sqrt{t}\}, \text{Pos}\{Y_1 = -\sqrt{t}\}\right\}, \end{aligned}$$

where $t \geq 0$, we obtain

$$\mu_{Y_1^2}(t) = \begin{cases} (1.6 + \sqrt{t})/4, & 0 \leq t \leq 2.4^2 \\ (6.4 - \sqrt{t})/4, & 2.4^2 \leq t \leq 6.4^2 \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Furthermore, we compute

$$\text{Cr}\{Y_1^2 \geq r\} = \begin{cases} (2 - \mu_{Y_1^2}(r))/2, & 0 \leq r \leq 2.4^2 \\ (\mu_{Y_1^2}(r))/2, & 2.4^2 \leq r \leq 6.4^2 \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Therefore, from Definition 1, we obtain $E[(X(V_1) - 7.4)^2] = E[Y_1^2]$ as follows:

$$\begin{aligned} E[Y_1^2] &= \int_0^\infty \text{Cr}\{Y_1^2 \geq r\} dr \\ &= \int_0^{2.4^2} \frac{1}{2} \left(2 - \frac{1.6 + \sqrt{r}}{4}\right) dr \\ &\quad + \int_{2.4^2}^{6.4^2} \frac{1}{2} \left(\frac{6.4 - \sqrt{r}}{4}\right) dr \\ &= 12.08. \end{aligned} \tag{7}$$

Similarly, we obtain $E[(X(V_2) - 7.4)^2] = E[Y_2^2] = 4.25$. Thus, $\text{Var}(X) = 0.2 \cdot E(X(V_1) - 7.4)^2 + 0.8 \cdot E(X(V_2) - 7.4)^2 = 0.2 \times 12.08 + 0.8 \times 4.25 = 5.81$.

3 Fuzzy Random Regression Model

Fuzzy arithmetic and fuzzy arithmetic operations for fuzzy numbers have been studied by making use of the extension principle [21], [22]. These studies have involved the concept of possibility. In 1984, Sanchez [26] discussed the solution of fuzzy equations in the same way as being intensively studied in the realm of fuzzy relational equations. Tanaka and Watada [32] pointed out that fuzzy equations discussed by Sanchez can be regarded as possibilistic equations.

In the sequel, possibilistic system has been applied to the linear regression analysis [29], [30], [39]. Our main concern here it to build a new fuzzy regression model for fuzzy random data, which is based on the possibilistic linear model.

Table I illustrates a format of data to be dealt with here, where input data X_{ik} and output data Y_i , for all $i = 1, \dots, N$ and $k = 1, \dots, K$ are fuzzy random variables, which are defined as

$$Y_i = \bigcup_{t=1}^{M_{Y_i}} \left\{ \left(Y_i^t, Y_i^{t,l}, Y_i^{t,r} \right)_T, p_i^t \right\}, \tag{8}$$

$$X_{ik} = \bigcup_{t=1}^{M_{X_{ik}}} \left\{ \left(X_{ik}^t, X_{ik}^{t,l}, X_{ik}^{t,r} \right)_T, q_{ik}^t \right\}, \tag{9}$$

respectively. This means that all values are given as fuzzy numbers with probabilities, where fuzzy variables $(Y_i^t, Y_i^{t,l}, Y_i^{t,r})_T$ and $(X_{ik}^t, X_{ik}^{t,l}, X_{ik}^{t,r})_T$ are associated with probability p_i^t and q_{ik}^t for $i = 1, 2, \dots, N$, $k = 1, 2, \dots, K$ and $t = 1, 2, \dots, M_{Y_i}$ or $t = 1, 2, \dots, M_{X_{ik}}$, respectively.

Let us denote fuzzy linear regression model as follows:

$$\bar{Y}_i = \bar{A}_1 X_{i1} + \dots + \bar{A}_K X_{iK}, \tag{10}$$

Table 1 Fuzzy random input-output data

Sample	Output	Inputs				
i	Y	X_1	X_2	\cdots	X_k	\cdots, X_K
1	Y_1	X_{11}	X_{12}	\cdots	X_{1k}	\cdots, X_{1K}
2	Y_2	X_{21}	X_{22}	\cdots	X_{2k}	\cdots, X_{2K}
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
i	Y_i	X_{i1}	X_{i2}	\cdots	X_{ik}	\cdots, X_{iK}
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
N	Y_N	X_{N1}	X_{N2}	\cdots	X_{Nk}	\cdots, X_{NK}

where \bar{Y}_i denotes an estimate of the output and $\bar{A}_k = \left(\frac{\bar{A}_k^l + \bar{A}_k^r}{2}, \bar{A}_k^l, \bar{A}_k^r \right)_T$ are symmetric triangular fuzzy coefficients when triangular fuzzy random data X_{ik} are given for $i = 1, \dots, N$ and $k = 1, \dots, K$ as shown in Table 1.

When outputs $Y_i = \bigcup_{t=1}^{M_{Y_i}} \left\{ (Y_i^t, Y_i^{t,l}, Y_i^{t,r})_T, p_i^t \right\}$, $i = 1, 2, \dots, N$ are given at the same time, we can determine the fuzzy random linear model so that the model includes all given fuzzy random outputs. Therefore, the following relation should hold:

$$\bar{Y}_i = \bar{A}_1 X_{i1} + \cdots + \bar{A}_K X_{iK} \underset{FR}{\supset} Y_i, \quad i = 1, \dots, N, \quad (11)$$

where $\underset{FR}{\supset}$ is a fuzzy random inclusion relation whose precise meaning will be explained later on. Following the principles of fuzzy arithmetic, the problem to obtain a fuzzy linear regression model results in the following mathematical programming problem:

[Regression model with fuzzy random data]

$$\left. \begin{aligned} \min_{\bar{A}} J(\bar{A}) &= \sum_{k=1}^K \left(\bar{A}_k^r - \bar{A}_k^l \right) \\ \text{subject to} & \\ \bar{A}_k^r &\geq \bar{A}_k^l, \\ Y_i &= \bar{A}_1 X_{i1} + \cdots + \bar{A}_K X_{iK} \underset{FR}{\supset} Y_i, \\ \text{for } i &= 1, \dots, N, k = 1, \dots, K. \end{aligned} \right\} \quad (12)$$

Here, the fuzzy random inclusion relation $\underset{FR}{\supset}$ is critical to the model (12), which can be defined in various ways. Watada and Wang [46] used the expectation-based inclusion, and converted the fuzzy random regression model (12) to the following expected value regression model which corresponds to the conventional fuzzy regression model:

[Fuzzy random expected value regression model]

$$\left. \begin{aligned}
 \min_{\bar{A}} J(\bar{A}) &= \sum_{k=1}^K \left(\bar{A}_k^r - \bar{A}_k^l \right) \\
 \text{subject to} & \\
 &\bar{A}_k^r \geq \bar{A}_k^l, \\
 &\bar{Y}_i = \sum_{k=1}^K \bar{A}_k E(X_{ik}) \underset{h}{\supseteq} E(Y_i), \\
 &\text{for } i = 1, \dots, N, k = 1, \dots, K,
 \end{aligned} \right\} \tag{13}$$

where $\underset{h}{\supseteq}$ denotes the fuzzy inclusion relation realized at level h .

In this study, we employ the confidence-interval based inclusion, which combines the expectation and variance of fuzzy random variables and the fuzzy inclusion relation satisfied at level h , to deal with the model (12). There are also some other ways to define the fuzzy random inclusion relation $\underset{FR}{\supseteq}$, which will yield more complicated fuzzy random regression models. For instance, in order to retain more complete information of the fuzzy random data, we can use the fuzzy inclusion relation directly for the product between a fuzzy parameter and a fuzzy value at some probability level. However, such calculation could be difficult since the product of two triangular fuzzy numbers does not retain the same triangular shape of the resulting membership function. Given this, the solution to the problem may rely on some heuristics as proposed in Watada and Pedrycz [45].

Table 2 Input-output data with confidence interval

Sample	Output	Inputs	
i	$I[e_Y, \sigma_Y]$	$I[e_{X_1}, \sigma_{X_1}]$	$\cdots I[e_{X_K}, \sigma_{X_K}]$
1	$I[e_{Y_1}, \sigma_{Y_1}]$	$I[e_{X_{11}}, \sigma_{X_{11}}]$	$\cdots I[e_{X_{1K}}, \sigma_{X_{1K}}]$
2	$I[e_{Y_2}, \sigma_{Y_2}]$	$I[e_{X_{21}}, \sigma_{X_{21}}]$	$\cdots I[e_{X_{2K}}, \sigma_{X_{2K}}]$
\vdots	\vdots	\vdots	\vdots
i	$I[e_{Y_i}, \sigma_{Y_i}]$	$I[e_{X_{i1}}, \sigma_{X_{i1}}]$	$\cdots I[e_{X_{iK}}, \sigma_{X_{iK}}]$
\vdots	\vdots	\vdots	\vdots
N	$I[e_{Y_N}, \sigma_{Y_N}]$	$I[e_{X_{N1}}, \sigma_{X_{N1}}]$	$\cdots I[e_{X_{NK}}, \sigma_{X_{NK}}]$

Before building the fuzzy random regression model, we define the confidence interval which is induced by the expectation and variance of a fuzzy random variable. When we consider the one sigma confidence ($1 \times \sigma$) interval of each fuzzy random variable, we can express it as the following interval

$$I[e_X, \sigma_X] \triangleq \left[E(X) - \sqrt{Var(X)}, E(X) + \sqrt{Var(X)} \right], \tag{14}$$

which is called a one-sigma confidence interval. Similarly, we can define two-sigma and three-sigma confidence intervals. All of these confidence intervals are called σ -confidence intervals. Table 2 shows the data with one-sigma confidence interval. Based on σ -confidence intervals, we formulate a new fuzzy random regression as follows:

[fuzzy random regression model (FRRM)]

$$\left. \begin{aligned} \min_{\bar{A}} J(\bar{A}) &= \sum_{k=1}^K \left(\bar{A}_k^r - \bar{A}_k^l \right) \\ \text{subject to} & \\ & \bar{A}_k^r \geq \bar{A}_k^l, \\ & \bar{Y}_i = \sum_{k=1}^K \bar{A}_k I[e_{X_{ik}}, \sigma_{X_{ik}}] \underset{h}{\supseteq} I[e_{Y_i}, \sigma_{Y_i}], \\ & \text{for } i = 1, \dots, N, k = 1, \dots, K. \end{aligned} \right\} \quad (15)$$

Remark 1. Given (15), if all the σ -confidence intervals of fuzzy random variables are considered as zero-sigma confidence intervals, the FRRM (15) will degenerate to the fuzzy random expected regression model described by (13).

Since the product of a fuzzy number (fuzzy coefficient) and an interval (confidence interval) is influenced by the signs of each component, to solve the FRRM (15), we need to take into account all the cases corresponding to different combinations of the signs of the fuzzy coefficients as well as the σ -confidence intervals of the fuzzy random data. The detailed computing associated with the the FRRM (15) will be discussed in the next section.

Table 3 Different cases of the product (19)

Case	Condition	Result
Case I	$\bar{e}_{ik} \geq \underline{e}_{ik} \geq 0$	
I-a	$\bar{a}_k \geq \underline{a}_k \geq 0$	$(\bar{A}_k \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}])_{h0} = [\underline{a}_k \cdot \underline{e}_{ik}, \bar{a}_k \cdot \bar{e}_{ik}]$
I-b	$\bar{a}_k \geq 0 \geq \underline{a}_k$	$(\bar{A}_k \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}])_{h0} = [\underline{a}_k \cdot \bar{e}_{ik}, \bar{a}_k \cdot \bar{e}_{ik}]$
I-c	$0 \geq \bar{a}_k \geq \underline{a}_k$	$(\bar{A}_k \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}])_{h0} = [\underline{a}_k \cdot \bar{e}_{ik}, \bar{a}_k \cdot \underline{e}_{ik}]$
Case II	$0 \geq \bar{e}_{ik} \geq \underline{e}_{ik}$	
II-a	$\bar{a}_k \geq \underline{a}_k \geq 0$	$(\bar{A}_k \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}])_{h0} = [\bar{a}_k \cdot \underline{e}_{ik}, \underline{a}_k \cdot \bar{e}_{ik}]$
II-b	$\underline{a}_k \leq 0 \leq \bar{a}_k$	$(\bar{A}_k \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}])_{h0} = [\bar{a}_k \cdot \underline{e}_{ik}, \underline{a}_k \cdot \underline{e}_{ik}]$
II-c	$0 \geq \bar{a}_k \geq \underline{a}_k$	$(\bar{A}_k \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}])_{h0} = [\bar{a}_k \cdot \bar{e}_{ik}, \underline{a}_k \cdot \underline{e}_{ik}]$
Case III	$\bar{e}_{ik} \geq 0 \geq \underline{e}_{ik}$	
III-a	$\bar{a}_k \geq \underline{a}_k \geq 0$	$(\bar{A}_k \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}])_{h0} = [\bar{a}_k \cdot \underline{e}_{ik}, \bar{a}_k \cdot \bar{e}_{ik}]$
III-b	$0 \geq \bar{a}_k \geq \underline{a}_k$	$(\bar{A}_k \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}])_{h0} = [\underline{a}_k \cdot \bar{e}_{ik}, \underline{a}_k \cdot \underline{e}_{ik}]$
III-c	$\bar{a}_k \geq 0 \geq \underline{a}_k$	$(\bar{A}_k \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}])_{h0} = [a_k^* \cdot e_{ik}^*, a_k^{**} \cdot e_{ik}^{**}]$

4 The Solution to the FRRM

The solution of FRRM (15) can be rewritten as a problem of N samples with one output and K input interval values. This problem is hard to solve, since it consists of $N \times K$ products between the fuzzy coefficients and confidence intervals. In order to solve the proposed FRRM, we can employ a vertices method as given below. That is, these multidimensional vertices are taken as new sample points with fuzzy output numbers. In the sequel, we can solve this problem using the conventional method. Nevertheless, this problem suffers from combinatorial explosion which becomes very much visible when the number of variables increases.

A heuristic method alleviates the difficulty to determine a fuzzy random regression model by using central values. This intends to simplify the calculations depending on many cases of the product between fuzzy numbers.

4.1 Vertices Method

Let us restructure the problem into that of N samples with one output interval value and 2^K vertices in K -dimension. That is, these multidimensional vertices are taken as new sample points with one output interval number. Therefore, we can solve this problem of $N \times 2^K$ samples with K values of input by the conventional method. Nevertheless, even in this section the problem suffers from the effect of combinatorial explosion.

Fuzzy random regression model can be developed to include the mean interval values of all samples in the model. Therefore, it is sufficient and necessary to consider only both two vertices of the end points of the interval of each dimension of a sample. For example, one sample with one input interval feature can be expressed with two vertices of the end points of the interval with a fuzzy output value. As a consequence, in FRRM (15), if we denote I_{ik}^L and I_{ik}^U left and right end points of the confidence intervals of the input X_{ik} , respectively, that is

$$I_{ik}^L = E(X_{ik}) - \sqrt{Var(X_{ik})}, \quad I_{ik}^U = E(X_{ik}) + \sqrt{Var(X_{ik})},$$

for $i = 1, 2, \dots, N, k = 1, 2, \dots, K$, the original fuzzy random regression model (15) can be converted into the following conventional fuzzy regression model by making use of the vertices method:

$$\left. \begin{aligned}
& \min_{\bar{A}} J(\bar{A}) = \sum_{k=1}^K (\bar{A}_k^r - \bar{A}_k^l) \\
& \text{subject to} \\
& \quad \bar{A}_k^r \geq \bar{A}_k^l, \\
& \quad (1) \rightarrow \bar{Y}_i = \bar{A}_1 \cdot I_{i1}^L + \bar{A}_2 \cdot I_{i2}^L + \cdots \\
& \quad \quad \quad + \bar{A}_K \cdot I_{iK}^L \underset{\sim}{\supseteq} I[e_{Y_i}, \sigma_{Y_i}], \\
& \quad (2) \rightarrow \bar{Y}_i = \bar{A}_1 \cdot I_{i1}^U + \bar{A}_2 \cdot I_{i2}^L + \cdots \\
& \quad \quad \quad + \bar{A}_K \cdot I_{iK}^L \underset{\sim}{\supseteq} I[e_{Y_i}, \sigma_{Y_i}], \\
& \quad (3) \rightarrow \bar{Y}_i = \bar{A}_1 \cdot I_{i1}^L + \bar{A}_2 \cdot I_{i2}^U + \cdots \\
& \quad \quad \quad + \bar{A}_K \cdot I_{iK}^L \underset{\sim}{\supseteq} I[e_{Y_i}, \sigma_{Y_i}], \\
& \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\
& \quad (2^K) \rightarrow \bar{Y}_i = \bar{A}_1 \cdot I_{i1}^U + \bar{A}_2 \cdot I_{i2}^U + \cdots \\
& \quad \quad \quad + \bar{A}_K \cdot I_{iK}^U \underset{\sim}{\supseteq} I[e_{Y_i}, \sigma_{Y_i}], \\
& \quad \text{for } i = 1, \dots, N, k = 1, \dots, K
\end{aligned} \right\} \quad (16)$$

where $\underset{h}{\supseteq}$ denotes the fuzzy inclusion relation realized at level h . The regression model (16) can be easily solved by taking advantage of the following linear programming provided that K is small:

$$\left. \begin{aligned}
& \min_{\bar{A}} J(\bar{A}) = \sum_{k=1}^K (\bar{A}_k^r - \bar{A}_k^l) \\
& \text{subject to} \\
& \quad \bar{A}_k^r \geq \bar{A}_k^l, \\
& \quad (1)^L \rightarrow \bar{Y}_i^l = \bar{A}_1^l \cdot I_{i1}^L + \bar{A}_2^l \cdot I_{i2}^L \\
& \quad \quad \quad + \cdots + \bar{A}_K^l \cdot I_{iK}^L \leq I_{Y_i}^L, \\
& \quad (1)^U \rightarrow \bar{Y}_i^r = \bar{A}_1^r \cdot I_{i1}^L + \bar{A}_2^r \cdot I_{i2}^L \\
& \quad \quad \quad + \cdots + \bar{A}_K^r \cdot I_{iK}^L \geq I_{Y_i}^U, \\
& \quad (2)^L \rightarrow \bar{Y}_i^l = \bar{A}_1^l \cdot I_{i1}^U + \bar{A}_2^l \cdot I_{i2}^L \\
& \quad \quad \quad + \cdots + \bar{A}_K^l \cdot I_{iK}^L \leq I_{Y_i}^L, \\
& \quad (2)^U \rightarrow \bar{Y}_i^r = \bar{A}_1^r \cdot I_{i1}^U + \bar{A}_2^r \cdot I_{i2}^L \\
& \quad \quad \quad + \cdots + \bar{A}_K^r \cdot I_{iK}^L \geq I_{Y_i}^U, \\
& \quad (3)^L \rightarrow \bar{Y}_i^l = \bar{A}_1^l \cdot I_{i1}^L + \bar{A}_2^l \cdot I_{i2}^U \\
& \quad \quad \quad + \cdots + \bar{A}_K^l \cdot I_{iK}^L \leq I_{Y_i}^L, \\
& \quad (3)^U \rightarrow \bar{Y}_i^r = \bar{A}_1^r \cdot I_{i1}^L + \bar{A}_2^r \cdot I_{i2}^U \\
& \quad \quad \quad + \cdots + \bar{A}_K^r \cdot I_{iK}^L \geq I_{Y_i}^U, \\
& \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\
& \quad (2^K)^L \rightarrow \bar{Y}_i^l = \bar{A}_1^l \cdot I_{i1}^U + \bar{A}_2^l \cdot I_{i2}^U \\
& \quad \quad \quad + \cdots + \bar{A}_K^l \cdot I_{iK}^U \leq I_{Y_i}^L, \\
& \quad (2^K)^U \rightarrow \bar{Y}_i^r = \bar{A}_1^r \cdot I_{i1}^U + \bar{A}_2^r \cdot I_{i2}^U \\
& \quad \quad \quad + \cdots + \bar{A}_K^r \cdot I_{iK}^U \geq I_{Y_i}^U, \\
& \quad \text{for } i = 1, \dots, N, k = 1, \dots, K.
\end{aligned} \right\} \quad (17)$$

In (17), there are 2^K inequalities for each sample i . Therefore, (17) has $K + 2 \times N \times 2^K$ inequalities in total, where $I_{Y_i}^L$ and $I_{Y_i}^U$ are the left and right end points of the confidence intervals of output Y_i , respectively.

Unfortunately this problem cannot be solved within a reasonable computing time when K becomes even moderately large. For example, when we have 1,000 features and 10,000 samples, the linear programming problem will come with $2 \times 10,000 \times 2^{1,000}$ constraints and 1,000 non-negative constraints. Given this, we have to resort to some heuristic strategies.

4.2 Heuristic Method

Let us use additional notations for variables $\bar{A}_k = [\underline{a}_k, \bar{a}_k]$ for $k = 1, 2, \dots, K$ in FRRM (15) and indicate step (n) of the algorithm by a suffix say, $\bar{A}_k^{(n)} = [\underline{a}_k^{(n)}, \bar{a}_k^{(n)}]$. According to the sign of A_k , $k = 1, 2, \dots, K$, the product of fuzzy number \bar{A}_k and $I[e_{X_{ik}}, \sigma_{X_{ik}}]$ involves three cases, for $i = 1, 2, \dots, N$. Let us use the notation $e_{ik} = E[X_{ik}]$, $\bar{e}_{ik} = E[X_{ik}] + Var[X_{ik}]$ and $\underline{e}_{ik} = E[X_{ik}] - Var[X_{ik}]$, respectively. Also, an α -level set of the fuzzy degree of a structural attribute $I[e_{X_{ik}}, \sigma_{X_{ik}}]$ ($i = 1, 2, \dots, N$, $k = 1, 2, \dots, K$) at the level h^0 is denoted as follows:

$$(\bar{A}_k)_{h^0} = [\underline{a}_k, \bar{a}_k]. \quad (18)$$

For each i and k , due to different combinations of signs of confidence interval $I[e_{X_{ik}}, \sigma_{X_{ik}}] = [\underline{e}_{ik}, \bar{e}_{ik}]$ and $(\bar{A}_k)_{h^0} = [\underline{a}_k, \bar{a}_k]$, the interval representing the product

$$(\bar{A}_k \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}])_{h^0} \quad (19)$$

requires several cases to be considered separately as covered in Table 3.

In Table 3 we have

$$a_k^* \cdot e_{ik}^* = \min \{ \underline{a}_k \cdot \bar{e}_{ik}, \bar{a}_k \cdot \underline{e}_{ik} \}$$

and

$$a_k^{**} \cdot e_{ik}^{**} = \max \{ \underline{a}_k \cdot \bar{e}_{ik}, \bar{a}_k \cdot \underline{e}_{ik} \},$$

respectively. As already underlined, it is difficult to derive analytical solutions to this problem and we resort ourselves to some heuristics. The proposed procedure can be outlined as follows (Algorithm 1 and Figure 1).

[Algorithm 1]

STEP 1. (Initial Setting)

The trial count n is set to 1, the termination count to N_{max} and using e_{ik} of attributes $k = 1, 2, \dots, K$ for each sample $i = 1, 2, \dots, N$, determine both the bounds $\underline{a}_k^{(n)}$ and $\bar{a}_k^{(n)}$ of $\bar{A}_k^{(n)}$ by solving the linear programming task(13).

STEP 2. (*LP Solution 1*)

Determine $\left(\bar{A}_k^{(n+1)} \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}]\right)_{h_0}$ according to Cases I-III, by assigning the signs of $\underline{a}_k^{(n)}$ and $\bar{a}_k^{(n)}$ to $\underline{a}_k^{(n+1)}$ and $\bar{a}_k^{(n+1)}$, for $k = 1, 2, \dots, K$. Determine $\underline{a}_k^{(n+1)}, \bar{a}_k^{(n+1)}$ for $k = 1, 2, \dots, K$ by the linear programming to minimize the fuzziness $J(\bar{A})$ under the constraints in the programming problem (15) according to the conditions for $\left(\bar{A}_i^{(n+1)} \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}]\right)_{h_0}$ as explained in Table 3.

STEP 3. (*Decision 1*)

If $\underline{a}_k^{(n+1)} \cdot \underline{a}_k^{(n)} \geq 0$ and $\bar{a}_k^{(n+1)} \cdot \bar{a}_k^{(n)} \geq 0$ ($k = 1, 2, \dots, K$) then go to **STEP 5**. Otherwise, let $n = n + 1$ and go to **STEP 4**

STEP 4. (*Decision 2*)

If the trial count n has not exceeded the given termination count N_{max} , then go to **STEP 2**. Otherwise, go to **STEP 5**.

STEP 5. (*Checking Vertices*)

Check all vertices whether each of them are included inside of the fuzzy regression lines. Then, place the vertex in set S_1 if it is included between the upper and lower lines, or on the upper or lower lines, of the obtained fuzzy regression; and place the vertex in set S_2 if it is outside the obtained fuzzy regression. If no vertex remains, then go to **STEP 6**; otherwise go to **STEP 5**.

STEP 6. (*Decision 3*)

If Set S_2 is null, then go to **STEP 8**. Otherwise go to **STEP 7**.

STEP 7. (*LP Solution 2*)

Adding all the outside points of fuzzy regression which are included S_2 to the constraints of the latest linear programming $LP^{(n)}$, and then resolve $LP^{(n)}$. Go to **STEP 8**.

STEP 8. (*Termination*)

Output the solution and terminate the algorithm.

Remark 2. Note that **STEP 1** solves the conventional fuzzy regression analysis in order to find the possible ranges of these coefficients. Using these signs we can solve the latter steps. In the linear programming (13), $E(X_{ik}) = e_{ik}$ for $i = 1, 2, \dots, N$ and $k = 1, 2, \dots, K$.

Remark 3. Note that we check the optimization status as outlined in **STEP 5**. The computational complexity of this process is only $O(2^K)$. When the algorithm finds the insufficient solution, we resolve the $LP^{(n)}$ using adding outlier vertices. The number of these vertices is very limited.

5 An Example

In this section, we present a simple example to visualize how to use the proposed FRRM. Assume that the fuzzy random input and output data (4 samples and 2 attributes) are given in the Tables 4 and 5, respectively.

Table 4 Input data

No.	X_1
1	$X_{11} = ((3, 2, 4)_T, 0.5; (4, 3, 5)_T, 0.5)$
2	$X_{21} = ((6, 4, 8)_T, 0.5; (8, 6, 10)_T, 0.5)$
3	$X_{31} = ((12, 10, 14)_T, 0.25; (14, 12, 16)_T, 0.75)$
4	$X_{41} = ((14, 12, 16)_T, 0.5; (16, 14, 18)_T, 0.5)$
No.	X_2
1	$X_{12} = ((2, 1, 3)_T, 0.1; (4, 3, 5)_T, 0.9)$
2	$X_{22} = ((3, 2, 4)_T, 0.5; (4, 3, 5)_T, 0.5)$
3	$X_{32} = ((12, 10, 16)_T, 0.2; (14, 12, 16)_T, 0.8)$
4	$X_{42} = ((18, 16, 20)_T, 0.2; (21, 20, 22)_T, 0.8)$

Table 5 Output data

No.	Y
1	$Y_1 = ((14, 10, 16)_T, 0.4; (18, 16, 20)_T, 0.6)$
2	$Y_2 = ((17, 16, 18)_T, 0.8; (20, 18, 22)_T, 0.2)$
3	$Y_3 = ((22, 20, 24)_T, 0.3; (26, 24, 28)_T, 0.7)$
4	$Y_4 = ((32, 30, 34)_T, 0.4; (36, 32, 40)_T, 0.6)$

The fuzzy regression model for the given data reads as follows:

$$\bar{Y}_i = \bar{A}_1 I[e_{X_{i1}}, \sigma_{X_{i1}}] + \bar{A}_2 I[e_{X_{i2}}, \sigma_{X_{i2}}],$$

where $I[e_{X_{ik}}, \sigma_{X_{ik}}]$ for $k = 1, 2$ are the one-sigma confidence intervals shown in (14). Since $N = 4, K = 2$, from the FRRM (15), taken $(\bar{A}_k)_{h^0} = [\bar{A}_k^l, \bar{A}_k^r], k = 1, 2$, the model can be built as

$$\left. \begin{aligned}
& \min_{\bar{A}} J(\bar{A}) = \bar{A}_1^r - \bar{A}_1^l + \bar{A}_2^r - \bar{A}_2^l \\
& \text{subject to} \\
& \bar{A}_1^r \geq \bar{A}_1^l, \bar{A}_2^r \geq \bar{A}_2^l \\
& \bar{Y}_1 = (\bar{A}_1)_{h^0} I[e_{X_{11}}, \sigma_{X_{11}}] \\
& \quad + (\bar{A}_2)_{h^0} I[e_{X_{12}}, \sigma_{X_{12}}] \underset{\sim}{\supseteq} I[e_{Y_1}, \sigma_{Y_1}], \\
& \bar{Y}_2 = (\bar{A}_1)_{h^0} I[e_{X_{21}}, \sigma_{X_{21}}] \\
& \quad + (\bar{A}_2)_{h^0} I[e_{X_{22}}, \sigma_{X_{22}}] \underset{\sim}{\supseteq} I[e_{Y_2}, \sigma_{Y_2}], \\
& \bar{Y}_3 = (\bar{A}_1)_{h^0} I[e_{X_{31}}, \sigma_{X_{31}}] \\
& \quad + (\bar{A}_2)_{h^0} I[e_{X_{32}}, \sigma_{X_{32}}] \underset{\sim}{\supseteq} I[e_{Y_3}, \sigma_{Y_3}], \\
& \bar{Y}_4 = (\bar{A}_1)_{h^0} I[e_{X_{41}}, \sigma_{X_{41}}] \\
& \quad + (\bar{A}_2)_{h^0} I[e_{X_{42}}, \sigma_{X_{42}}] \underset{\sim}{\supseteq} I[e_{Y_4}, \sigma_{Y_4}].
\end{aligned} \right\} \quad (20)$$

First of all, we need to calculate all the $I[e_{X_{ik}}, \sigma_{X_{ik}}]$ and $I[e_{Y_k}, \sigma_{Y_k}]$ for $i = 1, 2, 3, 4, k = 1, 2$. By using the calculation in Example 4, we obtain all the pairs $(e_{X_{ik}}, \sigma_{X_{ik}})$ and (e_{Y_k}, σ_{Y_k}) as shown in Table 6.

Table 6 Expectation and standard deviation of the data

i	$(e_{X_{i1}}, \sigma_{X_{i1}})$	$(e_{X_{i2}}, \sigma_{X_{i2}})$	(e_{Y_i}, σ_{Y_i})
1	(3.5, 0.56)	(3.8, 0.75)	(16.2, 7.68)
2	(7.0, 2.25)	(3.5, 0.56)	(17.6, 2.41)
3	(13.5, 1.87)	(13.7, 4.20)	(24.8, 4.68)
4	(15.0, 2.25)	(20.4, 2.00)	(34.4, 8.24)

Hence, the confidence intervals for the input data and output data can be calculated in the form

$$I[e_{X_{ki}}, \sigma_{X_{ki}}] = [e_{X_{ki}} - \sigma_{X_{ki}}, e_{X_{ki}} + \sigma_{X_{ki}}] \quad (21)$$

and

$$I[e_{Y_i}, \sigma_{Y_i}] = [e_{Y_i} - \sigma_{Y_i}, e_{Y_i} + \sigma_{Y_i}], \quad (22)$$

respectively, for $i = 1, 2$ and $k = 1, 2, 3, 4$. They are listed in the Tables 7 and 8, respectively.

Table 7 Confidence intervals of the input data

i	$I[e_{X_{i1}}, \sigma_{X_{i1}}]$	$I[e_{X_{i2}}, \sigma_{X_{i2}}]$
1	[2.94, 4.06]	[3.05, 4.75]
2	[4.75, 9.25]	[2.94, 4.06]
3	[11.63, 15.37]	[9.50, 17.90]
4	[12.75, 17.25]	[18.40, 22.40]

Table 8 Confidence intervals of the output data

i	$I[e_{Y_i}, \sigma_{Y_i}]$
1	[8.52, 23.88]
2	[15.19, 20.01]
3	[20.12, 29.48]
4	[26.16, 42.64]

Therefore, the model (20) can be written as

$$\left. \begin{aligned}
 & \min_{\bar{A}} J(\bar{A}) = \bar{A}_1^r - \bar{A}_1^l + \bar{A}_2^r - \bar{A}_2^l \\
 & \text{subject to} \\
 & \bar{A}_1^r \geq \bar{A}_1^l, \bar{A}_2^r \geq \bar{A}_2^l, \\
 & \bar{Y}_1 = [\bar{A}_1^l, \bar{A}_1^r] \cdot [2.94, 4.06] \\
 & \quad + [\bar{A}_2^l, \bar{A}_2^r] \cdot [3.05, 4.75] \supseteq [8.52, 23.88], \\
 & \bar{Y}_2 = [\bar{A}_1^l, \bar{A}_1^r] \cdot [4.75, 9.25] \\
 & \quad + [\bar{A}_2^l, \bar{A}_2^r] \cdot [2.94, 4.06] \supseteq [15.19, 20.01], \\
 & \bar{Y}_3 = [\bar{A}_1^l, \bar{A}_1^r] \cdot [11.63, 15.37] \\
 & \quad + [\bar{A}_2^l, \bar{A}_2^r] \cdot [9.5, 17.90] \supseteq [20.12, 29.48], \\
 & \bar{Y}_4 = [\bar{A}_1^l, \bar{A}_1^r] \cdot [12.75, 17.25] \\
 & \quad + [\bar{A}_2^l, \bar{A}_2^r] \cdot [18.40, 22.40] \supseteq [26.16, 42.64]
 \end{aligned} \right\} \quad (23)$$

We make use of Algorithm 1 to construct a regression model. Noting that $K = 2$, and all the confidence intervals $I[e_{X_{ik}}, \sigma_{X_{ik}}]$ are positive, from STEP 1 of the Algorithm 1, we need to set

$$\left(\bar{A}_k^{(1)} \cdot I[e_{X_{ik}}, \sigma_{X_{ik}}] \right)_{h_0} = \left[\underline{a}_k^{(1)} \cdot e_{ik}, \bar{a}_k^{(1)} \cdot e_{ik} \right],$$

for $i = 1, 2, 3, 4; k = 1, 2$, and the determine $\underline{a}_k^{(1)}$ and $\bar{a}_k^{(1)}$ for $k = 1, 2$,

Thus, the fuzzy random regression model is given in the form:

$$\begin{aligned}
 \bar{Y}_i &= \bar{A}_1 I[e_{X_{i1}}, \sigma_{X_{i1}}] + \bar{A}_2 I[e_{X_{i2}}, \sigma_{X_{i2}}] \\
 &= \bar{A}_1 I[e_{X_{i1}}, \sigma_{X_{i1}}] + \left(\frac{\bar{A}_2^l + \bar{A}_2^r}{2}, \bar{A}_2^l, \bar{A}_2^r \right)_T I[e_{X_{i2}}, \sigma_{X_{i2}}] \\
 &= 1.31 I[e_{X_{i1}}, \sigma_{X_{i1}}] + (3.29, 0.0, 6.57)_T I[e_{X_{i2}}, \sigma_{X_{i2}}].
 \end{aligned}$$

6 Concluding Remarks

In this chapter, we formulated a fuzzy random regression model, called FRRM (15) based on confidence intervals, by employing expectations and variances of fuzzy random variables being use here to construct the

σ -confidence intervals of fuzzy random data. The proposed model generalized our previous work, which dealt with a fuzzy random expected value regression model (13). Because it is difficult to determine the signs of arguments in calculations of the products of the fuzzy coefficients and confidence intervals, the proposed FRRM cannot be solved analytically.

Some solution approaches were discussed. The proposed vertices method can convert the original fuzzy random regression to a conventional fuzzy regression, which makes it possible to solve the original model by large-scale linear programming. However, the vertices method is limited by the size of the data, and so a heuristic algorithm was developed. This algorithm integrates linear programming and vertices checking, which enables us to handle the proposed regression by solving a series of linear programming problems. An illustrative example was provided to demonstrate the solution process.

These present work can be implemented in several application cases such as fuzzy random multi-attribute evaluation for production, and fuzzy random regression based evaluation of oil palm fruit grading. These applications will be discussed in our forthcoming studies.

References

1. Ammar, E.E.: On fuzzy random multiobjective quadratic programming. *European Journal of Operational Research* 193(2), 530–539 (2009)
2. Chang, Y.-H.O.: Hybrid fuzzy least-squares regression analysis and its reliability measures. *Fuzzy Sets and Systems* 119(2), 225–246 (2001)
3. Chang, P., Lin, K., Lin, C., Hung, K., Hung, L., Hsu, B.: Developing a fuzzy bi-cluster regression to estimate heat tolerance in plants by chlorophyll fluorescence. *IEEE Transactions on Fuzzy Systems*, doi:10.1109/TFUZZ.2008.924349, forth coming article
4. Choi, S.H., Buckley, J.J.: Fuzzy regression using least absolute deviation estimators. *Soft Computing* 12(3), 257–263 (2008)
5. Hao, P.-Y., Chiang, J.-H.: Fuzzy Regression Analysis by Support Vector Learning Approach. *IEEE Transactions on Fuzzy Systems* 16(2), 428–441 (2008)
6. Hop, N.V.: Fuzzy stochastic goal programming problems. *European Journal of Operational Research* 176(1), 77–86 (2007)
7. Hong, Y.-Y., Chao, Z.-T.: Development of energy loss formula for distribution systems using FCN algorithm and cluster-wise fuzzy regression. *IEEE Transactions on Power Delivery* 17(3), 794–799 (2002)
8. Hong, D.H., Hwang, C.H.: Interval regression analysis using quadratic loss support vector machine. *IEEE Transactions on Fuzzy Systems* 13(2), 229–237 (2005)
9. Imoto, S., Yabuuchi, Y., Watada, J.: Fuzzy regression model of R&D project evaluation. *Applied Soft Computing* 3(6), 1–7 (2007)
10. Kandel, A., Byatt, W.J.: Fuzzy sets, fuzzy algebra, and fuzzy statistics. *Proceedings of the IEEE* 66(12), 1619–1639 (1978)
11. Kruse, R., Meyer, K.D.: *Statistics with Vague Data*. D. Reidel Publishing Company, Dordrecht (1987)

12. Kwakernaak, H.: Fuzzy random variables–I. Definitions and theorems. *Information Sciences* 15(1), 1–29 (1978)
13. Liu, B.: Fuzzy random chance-constrained programming. *IEEE Transactions on Fuzzy Systems* 9(5), 713–720 (2001)
14. Liu, B., Liu, Y.-K.: Expected value of fuzzy variable and fuzzy expected value models. *IEEE Transactions on Fuzzy Systems* 10(4), 445–450 (2002)
15. Liu, Y.-K., Liu, B.: Fuzzy random variable: A scalar expected value operator. *Fuzzy Optimization and Decision Making* 2(2), 143–160 (2003)
16. Liu, Y.-K.: The approximation method for two-stage fuzzy random programming with recourse. *IEEE Transactions on Fuzzy Systems* 15(6), 1197–1208 (2007)
17. López-Díaz, M., Gil, A.: Constructive definitions of fuzzy random variables. *Statistics and Probability Letters* 36(2), 135–143 (1997)
18. Luhandjula, M.K.: Fuzziness and randomness in an optimization framework. *Fuzzy Sets and Systems* 77(3), 291–297 (1996)
19. Nahmias, S.: Fuzzy variables. *Fuzzy Sets and Systems* 1(2), 97–111 (1978)
20. Nazarko, J., Zalewski, W.: The fuzzy regression approach to peak load estimation in power distribution systems. *IEEE Transactions on Power Systems* 14(3), 809–814 (1999)
21. Negoita, C.V., Ralescu, D.A.: *Application of Fuzzy Sets to Systems Analysis*. Birkhäuser, Basel (1975)
22. Nguyen, H.T.: A note on the extension principle for fuzzy sets. *Journal of Mathematical Analysis and Applications* 64(2), 369–380 (1978)
23. Nguyen, H.T., Wu, B.: *Fundamentals of Statistics with Fuzzy Data*. Springer, New York (2006)
24. Niimura, T., Nakashima, T.: Deregulated electricity market data representation by fuzzy regression models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 31(3), 320–326 (2001)
25. Puri, M.L., Ralescu, D.A.: Fuzzy random variables. *Journal of Mathematical Analysis and Applications* 114(2), 409–422 (1986)
26. Sanchez, E.: Solution of fuzzy equations with extended operations. *Fuzzy Sets and Systems* 12(3), 237–248 (1984)
27. Song, K.-B., Baek, Y.-S., Hong, D.H., Jang, G.: Short-term load forecasting for the holidays using fuzzy linear regression method. *IEEE Transactions on Power Systems* 20(1), 96–101 (2005)
28. Sun, C.M., Wu, B.: New statistical approaches for fuzzy data. *International Journal of Uncertainty, Fuzziness & Knowledge-Based Systems* 15(2), 89–106 (2007)
29. Tanaka, H., Uejima, S., Asai, K.: Linear regression analysis with fuzzy model. *IEEE Transactions on Systems, Man and Cybernetics* 12(6), 903–907 (1982)
30. Tanaka, H., Shimomura, T., Watada, J., Asai, K.: Fuzzy linear regression analysis of the number of staff in local government. In: *Proceedings of FIP 1984, Kauai, Hawaii, July 22–26 (1984)*
31. Tanaka, H., Hayashi, I., Watada, J.: Possibilistic linear regression for fuzzy data. *European Journal of Operational Research* 40(3), 389–396 (1989)
32. Tanaka, H., Watada, J.: Possibilistic linear systems and their application to the linear regression model. *Fuzzy Sets and Systems* 27(3), 275–289 (1988)
33. Tanaka, H., Lee, H.: Interval regression analysis by quadratic programming approach. *IEEE Transactions on Fuzzy Systems* 6(4), 473–481 (1998)

34. Toyoura, Y., Watada, J., Khalid, M., Yusof, R.: Formulation of linguistic regression model based on natural words. *Soft Computing* 8(10), 681–688 (2004)
35. Wang, S., Watada, J.: Reliability optimization of a series-parallel system with fuzzy random lifetimes. *International Journal of Innovative Computing, Information & Control* 5(6), 1547–1558 (2009)
36. Wang, S., Liu, Y.-K., Watada, J.: Fuzzy random renewal process with queueing applications. *Computers & Mathematics with Applications* 57(7), 1232–1248 (2009)
37. Wang, G., Qiao, Z.: Linear programming with fuzzy random variable coefficients. *Fuzzy Sets and Systems* 57(3), 295–311 (1993)
38. Watada, J., Tanaka, H.: The perspective of possibility theory in decision making. In: Sawaragi, Y., Inoue, K., Nakayama, H. (eds.) VII-th Int. Conf. Post Conference Book, Multiple Criteria Decision Making - Toward Interactive Intelligent Decision Support Systems, pp. 328–337. Springer, Heidelberg (1986)
39. Watada, J., Tanaka, H., Asai, K.: Analysis of time-series data by possibilistic model. In: *Proceedings of International Workshop on Fuzzy System Applications*, Fukuoka, pp. 228–233 (1988)
40. Watada, J.: Fuzzy time-series analysis and forecasting of sales volume. In: Kacprzyk, J., Fedrizzi, M. (eds.) *Fuzzy Regression Analysis*, pp. 211–227. Omnitel Press, Warsaw (1992)
41. Watada, J., Tanaka, H., Asai, K.: Fuzzy quantification theory: Type I. *Official Journal of The Behaviormetrics Society of Japan (Behaviormetrics)* 11(1), 66–73 (1984) (in Japanese)
42. Watada, J.: Possibilistic time-series analysis and its analysis of consumption. In: Dubois, D., Yager, M.M. (eds.) *Fuzzy Information Engineering*, pp. 187–200. John Wiley Sons, Inc., Chichester (1996)
43. Watada, J., Mizunuma, H.: Fuzzy switching regression model based on genetic algorithm. In: *Proceedings of the 7th International Fuzzy Systems Association World Congress (IFSA 1997)*, Prague, Czech Republic, pp. 113–118 (1997)
44. Watada, J., Toyoura, Y.: Formulation of Fuzzy Switching Auto-Regression Model. *International Journal of Chaos Theory and Applications* 7(1-2), 67–76 (2002)
45. Watada, J., Pedrycz, W.: A fuzzy regression approach to acquisition of linguistic rules. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) *Handbook on Granular Computation*, ch. 32, pp. 719–740. John Wiley & Sons Ltd., Chichester (2008)
46. Watada, J., Wang, S.: Regression model based on fuzzy random variables. In: Rodulph, S. (ed.) *Views on Fuzzy Sets and Systems from Different Perspectives*, ch. 26. Springer, Berlin (2009)
47. Watada, J., Wang, S., Pedrycz, W.: Building confidence-interval-based fuzzy random regression models. *IEEE Trans. Fuzzy Systems* 17(6), 1273–1283 (2009)
48. Yin, S.A., Chang, R.F., Lu, C.N.: Reliability worth assessment of high-tech industry. *IEEE Transactions on Power Engineering Review* 22(11), 56–56 (2002)
49. Zadeh, L.A.: Fuzzy probabilities. *Information Processing & Management* 20(3), 363–372 (1984); *Information Science* 172(1-2), 1–40 (2005); *Computational Statistics and Data Analysis* 51(1), 15–46 (2006)

Evolutionary Multiobjective Neural Network Models Identification: Evolving Task-Optimised Models

Pedro M. Ferreira and António E. Ruano

Abstract. In the system identification context, neural networks are *black-box* models, meaning that both their parameters and structure need to be determined from data. Their identification is often done iteratively in an *ad-hoc* fashion focusing the first aspect. Frequently the selection of inputs, model structure, and model order are overlooked subjects by practitioners, because the number of possibilities is commonly huge, thus leaving the designer at the hands of the *curse of dimensionality*. Moreover, the design criteria may include multiple conflicting objectives, which gives to the model identification problem a multiobjective combinatorial optimisation character. *Evolutionary multiobjective optimisation algorithms* are particularly well suited to address this problem because they can evolve optimised model structures that meet pre-specified design criteria in acceptable computing time. In this article the subject is reviewed, the authors present their approach to the problem in the context of identifying neural network models for time-series prediction and for classification purposes, and two application case studies are described, one in each of these fields.

1 Introduction

In most practical applications of Artificial Neural Networks (ANN), they are used to perform a non-linear mapping between an input space, \mathbf{X} , and an output space, \mathbf{y} , in order to model complex relationships between these or to detect patterns in

Pedro M. Ferreira

Algarve Science & Technology Park, University of Algarve,
Campus de Gambelas - Pav. A5, 8005-139 Faro, Portugal
e-mail: pfrazao@ualg.pt

António E. Ruano · Pedro M. Ferreira

Centre for Intelligent Systems, University of Algarve - FCT,
Campus de Gambelas, 8005-139 Faro, Portugal
e-mail: aruano@ualg.pt

input-output data. These functionalities correspond mostly to function approximation problems in the context of static or dynamic models identification, or decision problems in the contexts of pattern matching and classification. The non-linear mapping function illustrated in Fig. 1 is given by:

$$\hat{y}_k = g(\mathbf{x}_k, \mathbf{w}) \quad (1)$$

Usually, given a data set $\mathbf{D} = (\mathbf{X}, \mathbf{y})$ composed of N input-output pairs, the network

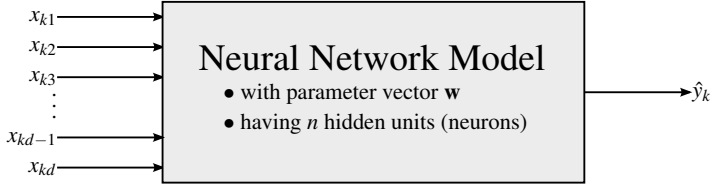


Fig. 1 Illustration of a general mapping $\hat{y}_k = g(\mathbf{x}_k, \mathbf{w})$

parameter vector \mathbf{w} is computed in order to minimise the sum-of-squares of the mapping error, i.e.,

$$\varepsilon = \mathbf{e}^T \mathbf{e} \quad (2)$$

where,

$$\begin{aligned} \mathbf{e} &= \mathbf{y} - \hat{\mathbf{y}}, & e_k &= y_k - \hat{y}_k, \\ \hat{\mathbf{y}} &= g(\mathbf{X}, \mathbf{w}), & \hat{y}_k &= g(\mathbf{x}_k, \mathbf{w}), \\ \mathbf{X} &= [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T, & \mathbf{y} &= [y_1, y_2, \dots, y_N]^T, \\ \mathbf{x}_k &= [x_{k1}, x_{k2}, \dots, x_{kd}]. \end{aligned} \quad (3)$$

In many applications the set of d input features, x_{ki} , needs to be selected from a larger set, \mathbf{F} , often having a dimension significantly larger than a prescribed maximum input vector dimension, d_M . Assuming \mathbf{F} has q features it may be specified as,

$$\begin{aligned} \mathbf{F} &= [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_q], \\ \mathbf{f}_l &= [f_{1l}, f_{2l}, \dots, f_{Nl}]^T. \end{aligned} \quad (4)$$

Then, the input data set \mathbf{X} is constructed by selecting d columns from \mathbf{F} such that,

$$\begin{aligned} \mathbf{x}_k &= [x_{k1}, x_{k2}, \dots, x_{kd}] = \\ &= [f_{k\lambda_1}, f_{k\lambda_2}, \dots, f_{k\lambda_d}], \end{aligned} \quad (5)$$

where the λ_j are indices to the columns of \mathbf{F} .

Also, depending on the type of ANN to be employed, the number n of hidden units (artificial neurons or simply neurons) must be specified. Once d input features

are selected and the number of neurons, n , is specified, the ANN parameter vector, \mathbf{w} , is computed by means of a suitable training algorithm.

By taking into account these three aspects of the ANN model identification process, the problem addressed in this article may be generally stated as follows:

Considering the application at hands, select d ($d_m < d \leq d_M$) input features from the set \mathbf{F} , a suitable number of neurons n ($n_m < n \leq n_M$), and compute the ANN parameter vector \mathbf{w} , such that the *best* ANN mapping, given in (1), is obtained.

It is already clear that the ANN design problem may be separated in two distinct sub-problems, each reflecting different aspects of the design:

- ANN parameters relates to the network parameters. Includes their computation by means of a training algorithm.
- ANN structure relates to the network topology. Includes the selection of suitable inputs and an appropriate number of neurons;

Many techniques have been proposed to solve both sub-problems, either separately or jointly, some failing to capture their distinct nature, therefore not fully exploiting existing approaches that are considered more appropriate. The first is a non-linear parameter optimization problem, to which non-linear gradient-based methods have proven to be superior. The second is a combinatorial optimisation problem that, as will be shown, needs to be addressed from a multiobjective optimisation perspective.

In Sect. 2 a more precise definition of the problem statement will be given and the approach followed successfully by the authors in a number of applications will be presented. From these, two were selected and are described in Sect. 3, one in the field of time-series modelling and forecasting, the other in the area of classification problems. The results from the application of the methodology to the selected ANN design problems will be presented and discussed in Sects. 3.1.2 and 3.2.2, respectively for each design problem. Finally, some concluding remarks will be made in Sect. 4.

2 Methodology

The problem statement presented in the previous section is of a general nature leaving open two vague notions that need to be elaborated in order to provide a formal problem definition. On one hand the concept of "best ANN mapping" requires the definition of *best*. On the other, the sentence "considering the application at hands" implies that the problem will be solved by taking the application into account. In fact the two notions are related as it seems appropriate to define what is a "best ANN, considering the application at hands".

2.1 Problem Definition

In order to define a "best ANN, considering the application at hands", one or more quality measures are required so that any two different ANN solutions may be compared and a decision can be reached on which is best. The problem decomposition given in Sect. 1 suggests the existence of quality measures for each sub-problem and gives clear hints on how to choose them:

ANN parameters	The quality measures should reflect how well did the training stage performed and how good is the mapping obtained by the parameters computed.
ANN structure	The quality measures should tell how fit is the ANN structure for the application at hands.

This breakdown in the nature of the quality measures allows the definition of a two component quality vector as,

$$\begin{aligned} \mu(\mathbf{F}, \Lambda, n, \mathbf{w}) &= [\mu^p, \mu^s], \\ \mu^p &= [\mu_1^p, \mu_2^p, \dots, \mu_u^p], \\ \mu^s &= [\mu_1^s, \mu_2^s, \dots, \mu_v^s], \\ \Lambda &= [\lambda_1, \lambda_2, \dots, \lambda_d], \end{aligned} \quad (6)$$

where μ^p and μ^s contain u and v quality measures related to each of the sub-problems, Λ is the vector of indices to the columns of \mathbf{F} that defines the input features considered, and the superscripts p and s denote quality measures related to the ANN training stage and to the ANN fitness for the specific application, respectively. The dependence on \mathbf{F} , Λ , n , and \mathbf{w} , has been made explicit only for μ for easiness of reading.

Assuming that the quality measures in $\mu(\mathbf{F}, \Lambda, n, \mathbf{w})$ are well defined quantities specifying objective functions that should be minimised in order to obtain the "best ANN for the application at hands", the problem statement given in Sect 1 may now be formally defined as:

Select $d \in [d_m, d_M]$ input features from \mathbf{F} , $n \in [n_m, n_M]$ neurons, and compute \mathbf{w} , such that $\mu(\mathbf{F}, \Lambda, n, \mathbf{w})$ is minimised. Formally,

$$\begin{aligned} \min_{\Lambda, n, \mathbf{w}} \mu(\mathbf{F}, \Lambda, n, \mathbf{w}), \text{ given:} \\ (\mathbf{F}, \mathbf{y}), \\ d \in [d_m, d_M], \\ n \in [n_m, n_M]. \end{aligned} \quad (7)$$

Given the definition of $\mu(\mathbf{F}, \Lambda, n, \mathbf{w})$ (simply μ in the following), it is likely that some objective functions are conflicting, e.g. in μ^p Eq. 2 may be minimised and in

μ^s there could be an objective to minimise the complexity of the ANN, expressing the goals of improving performance while decreasing the network size. Therefore the search problem defined in (7) is a combinatorial multiobjective optimisation problem which does not have a single solution minimising all components of μ simultaneously. Instead, the solution is the set of Pareto points in search space (or design space) that define the Pareto front in the space of objectives. This means that the ANN model designer has to select one particular ANN by examining trade-offs in the objectives of the Pareto front.

Searching exhaustively over the search space defined by $(\mathbf{F}, [d_m, d_M], [n_m, n_M])$ is the preferred solution as it allows finding the true Pareto front, but this is normally unfeasible in useful time due to the complexity of evaluating μ and to the size of the search space. Although trial and error may provide an approach to guide the search, the number of possibilities is often enormous and it may result in the execution of many trials without obtaining acceptable objective values in μ . Moreover, the results from the trials may easily misguide the designer into some poor local minima as the relation between search space and objective space is unknown.

Although a good number of techniques have been proposed over the years to deal with multiobjective problems it was more recently that the potential of Evolutionary algorithms (EAs) to approximate the Pareto front was recognised, generating a research area now known as evolutionary multiobjective optimisation (EMO). Multiobjective evolutionary algorithms (MOEAs) have proven to be robust and efficient when dealing with problems with multiple conflicting objectives and with very large and complex search spaces, therefore they are employed here to solve the ANN structure search problem. A review about the EMO field and MOEAs is beyond the scope of this article, the interested reader can find detailed descriptions in textbooks, e.g. [10], and excellent overviews on [48, 7, 8].

The application of EAs to the design of ANN models has been addressed by many researchers, with variations on the aspects of ANN design that are considered. Distinct formulations employ EAs in order to optimise/select: the number of neurons and network parameters [32, 5, 3, 6, 47, 29]; both the topology and parameters [28, 25]; the complete topology [27, 2]; or, only the network inputs [34]. A discussion considering the different possible formulations may be found in [4]. The approach herein presented follows previous work [21, 36] in the context of polynomial models identification. It has been applied by the authors in the contexts of time-series modelling and prediction [43, 16, 17, 41, 14, 37, 18, 42], and classification [9, 12].

2.2 Multiobjective Evolutionary Algorithms

MOEAs are one class of EAs that benefit from a set of procedures and operators inspired on the process of natural evolution and on the notion of survival of the fittest, in order to perform a population based search for the Pareto set of solutions of a given multiobjective problem. The solution candidates are called *individuals* and their set is referred to as the *population*. One run of a MOEA starts with an

initial population of individuals, the initial *generation*, which are then evaluated and manipulated to compute the population of individuals composing the next generation. Hopefully, after a sufficient number of generations the population has evolved achieving a satisfactory approximation to the Pareto front.

The operation of most MOEAs follows the flow illustrated in Fig. 2 where the main procedures and operators are shown. At each iteration the population is

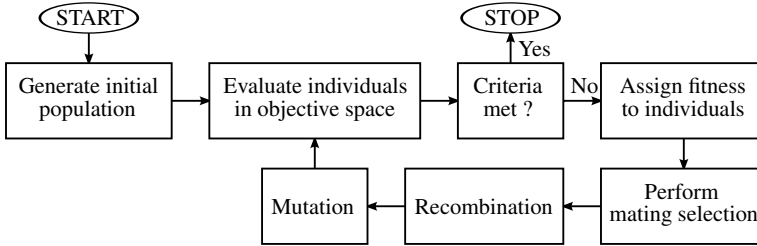


Fig. 2 Typical flow of operation of most MOEAs

evaluated for the objectives specified in μ and a check is made to ascertain if the design criteria was met. If this is the case the MOEA stops and the designer obtains the individuals that form the current approximation to the Pareto front, otherwise the algorithm proceeds. In this case each individual in the population is assigned a fitness value and based on this fitness the individuals are mated. Afterwards each mated pair will produce two offspring by the application of the recombination operator, thus forming the next generation. Finally the mutation operator is applied to each children before repeating the whole process.

2.2.1 Individual Representation Each individual in the MOEA population must be specified by a representation, the *chromosome*, encoding the topology of an ANN. Most frequently, *feed-forward* ANNs are employed in modelling, prediction and classification problems, usually having one or two hidden layers of neurons. In the following the general class of feed-forward ANNs having one hidden layer of neurons is considered. As will be shown, if two or more layers are used only slight changes are required in the chromosome and in the mutation operator.

The topology of the ANN architectures just mentioned may be completely specified by the number of neurons n and by the indices Λ to the columns of \mathbf{F} , defining the input features to be employed. Therefore the chromosome is a string of integers, the first representing the number of neurons and the remaining representing the subset of input terms taken from \mathbf{F} . The chromosome definition is shown in Fig. 3. The multiobjective optimisation problem defined in (7) states that the number of inputs d is required to be in the range $[d_m, d_M]$. This corresponds to a variable length chromosome having at least d_m input terms. The first component corresponds to the number of neurons, those highlighted by a light grey background represent the minimum number of inputs, and the remaining are a variable number of input terms up

Chromosome:

n	λ_1	λ_2	\dots	λ_{d_m}	λ_{d_m+1}	\dots	λ_{d_M}
-----	-------------	-------------	---------	-----------------	-------------------	---------	-----------------

Input space of q features, \mathbf{F} :

\mathbf{f}_1	\mathbf{f}_2	\dots	\mathbf{f}_{a_0}	\mathbf{f}_{a_0+1}	\mathbf{f}_{a_0+2}	\dots	$\mathbf{f}_{a_0+a_1}$	\dots	$\mathbf{f}_{a_0+\dots+a_o}$	\mathbf{f}_q	
$y(t)$	$y(t-1)$	\dots	$y(t-\tau_y)$	$v_1(t)$	$v_1(t-1)$	\dots	$v_1(t-\tau_{v_1})$	\dots	$v_o(t)$	\dots	$v_o(t-\tau_{v_o})$

Fig. 3 Chromosome and input space lookup table

to (in total) d_M . The λ_j values are the indices of the features \mathbf{f}_l in the columns of \mathbf{F} . In cases where the ANN acts as a predictor the input-output structure is, in the most general form, given by a non-linear autoregressive (NAR) with exogenous inputs (NARX),

$$\begin{aligned}
 y(t+1) &= g(y(t), y(t-1), \dots, y(t-\tau_y), \\
 &\quad v_1(t), v_1(t-1), \dots, v_1(t-\tau_{v_1}), \\
 &\quad \dots, \\
 &\quad v_o(t), v_o(t-1), \dots, v_o(t-\tau_{v_o})),
 \end{aligned} \tag{8}$$

where y is the output and v_1 to v_o are o exogenous inputs. In such cases \mathbf{F} is composed of a_0 output delayed terms with a maximum lag of τ_y and a_i input terms for each exogenous variable v_i , each having τ_{v_i} as maximum lag. The correspondence between the features in the columns of \mathbf{F} and the inputs of a NARX model is depicted in the lower part of Fig. 3, where the inputs corresponding to delayed output values are highlighted by a light grey background.

It should be noted that the chromosome would require a small change if the ANN considered had multiple hidden layers. In this situation as many additional components as the number of additional hidden layers would be inserted at the beginning of the chromosome, in order to encode the number of neurons in the various layers.

2.2.2 MOEAs Procedures and Operators Once evaluated in objective space each individual is assigned a scalar value, the *fitness*, that should reflect that individual's quality. The fitness assignment strategy is one of the distinguishing characteristics of existing MOEAs, thus is usually dependant on the MOEA used in practice. In general these strategies are based on different principles and belong to one of three classes: aggregation based; criterion based; and Pareto based strategies. For more detailed discussions on these strategies, the reader should consult the literature on the MOEA being used or one of the references given above about MOEAs.

The mating procedure uses the population fitness information in order to create a *mating pool* with pairs of individuals that will be combined to form the basis of the next generation population. It is commonly implemented as a sampling procedure where the individuals having higher fitness have increased chance of getting multiple copies in the mating pool, and those with lower fitness have little or no chance

of getting there. The result is that the fittest individuals have a higher probability of breeding as opposed to the worse individuals that are unlikely to influence the new generation.

With a given probability, the *crossover probability*, every pair of individuals in the mating pool produces two offspring by exchanging part of their chromosomes. This is accomplished by the recombination operator, whose operation is illustrated in Fig. 4 by splitting the procedure in two steps. First, the chromosomes are re-ordered, secondly, parts of the chromosomes are exchanged. Reordering is also

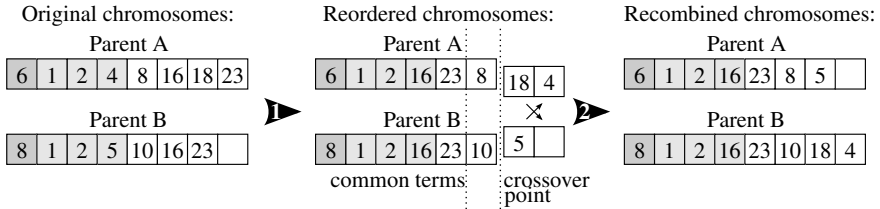


Fig. 4 Crossover recombination operator

accomplished in two steps: common terms in the chromosomes are swapped to the left-most positions, then the remaining terms are shuffled. This way the common terms in the chromosomes are isolated in a way that makes them unavailable for the exchange. A point is then randomly chosen, the *crossover point*, and the elements to its right are exchanged. This procedure, known as *full identity preserving crossover* [21, 24], guarantees offspring with no duplicate terms.

Mutation is applied to the new population generated after recombination, independently in two parts of the chromosome. The number(s) of neurons in the hidden layer(s) of the ANN are mutated with a given probability by adding or subtracting one neuron to the existing quantity. Care must be taken in order to guarantee the boundary conditions $n_m \leq n \leq n_M$. The input terms are mutated with a given probability by one of three operations: replacement, addition or deletion. First, each term is tested and is either deleted or replaced by another term from the set of those outside the chromosome. Deletion only occurs if the chromosome has more terms than the minimum specified, d_m . After this, if the chromosome is not full, one term may be added by selecting it from the set of those outside the chromosome.

After completing the operations described above the MOEA flow proceeds to the evaluation step and the cycle repeats itself for the new population of individuals. Therefore, some criterion is required to stop the MOEA execution. The most simple approach consists in stopping the execution after a predefined number of generations. Other options include testing the objectives and design criteria and stop the execution if a satisfactory individual is found, or checking the population for stagnation. The latter option may be accomplished, for instance, by specifying a maximum number of consecutive generations during which no change is observed in the Pareto front approximation.

2.3 Model Design Cycle

Globally, the ANN structure optimisation problem can be viewed as sequence of actions undertaken by the model designer, which should be repeated until pre-specified design goals are achieved. These actions can be grouped into three major categories: problem definition, solution(s) generation and analysis of results. In the context of this identification framework, the procedure is executed as depicted in Fig. 5. In

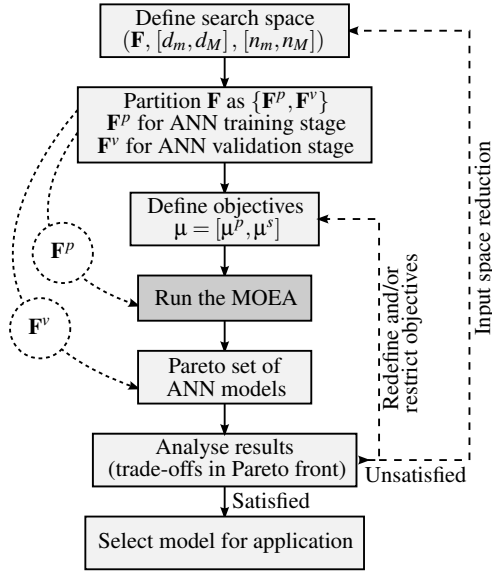


Fig. 5 Model design cycle

summary, the problem definition is carried out by choosing a number of hypothetically relevant input features to construct \mathbf{F} , by specifying the minimum and maximum size of the chromosome input terms string, and by defining the range allowed for the number of neurons of the ANNs. In the case of a NARX predictive model identification, the specification of \mathbf{F} corresponds to the selection of input variables and the corresponding lags considered. This stage affects the size of the search space. The input search space is then partitioned into two data sets, \mathbf{F}^p and \mathbf{F}^v , the first intended for the ANN parameter training procedure, the second to validate the results obtained by the Pareto set of individuals. The validation step serves the purpose of detecting any bias that may have occurred towards the \mathbf{F}^p data set during the MOEA model structure optimisation.

Another aspect to be defined is the set of objectives and goals to be attained. The objectives specified in μ^s play an important role in the adequacy of the models obtained to the application problem being considered. Therefore they should be designed to express the quality of an individual in the context of the final application. Specifying μ affects the quantity, quality and class of the resulting solutions.

When the analysis of the solutions provided by the MOEA requires the process to be repeated, the problem definition steps should be revised. In this case, two major actions can be carried out: input space redefinition by removing or adding one or more features (variables and lagged input terms in the case of modelling problems), and improving the trade-off surface coverage by changing objectives or redefining goals. This process may be advantageous as usually the output of one run allows reducing the number of input terms (and possibly variables for modelling problems) by eliminating those not present in the resulting population. Also, it usually becomes possible to narrow the range for the number of neurons in face of the results obtained in one run. This results in a smaller search space in a subsequent run of the MOEA, possibly achieving a faster convergence and better approximation of the Pareto front. This cycle of actions can be iterated until a refined set of satisfactory solutions is obtained.

2.4 ANN Parameter Training

In Sects. 1 and 2.1, the ANN identification problem has been decomposed in two sub-problems, the first one, related to the optimisation of the network parameters, being usually treated as a non-linear optimisation problem. It is clear that the training procedure is most often dependant on the specific ANN being employed, although some procedures may easily be adapted to various kinds of ANNs. The class of feed-forward ANNs include, among others, radial basis function (RBF) networks, multi-layer perceptrons (MLPs), B-spline networks, wavelet networks, and some types of neuro-fuzzy networks. Importantly, a common topology of these architectures share the property of *parameter separation*, i.e., they can be regarded as a non-linear/linear topology because one or more hidden layers of non-linear neurons are followed by a linear combination of neuron outputs to produce the network overall result. It is commonly accepted that gradient-based algorithms, in particular the Levenberg-Marquardt (LM) algorithm [31], outperforms other parameter training methods, and it has been shown that methods exploiting the separability of parameters [39, 40, 13] achieve increased accuracy and convergence rates.

The two example ANN identification problems that will be introduced in Sect. 3 employ RBF neural networks (NNs) and the LM algorithm in the minimisation of a modified training criterion that exploits the separability of parameters as found in RBF NNs. For this reason an outline of the training procedure is given in the following sub-sections.

2.4.1 Training Criterion For simplicity, but without loss of generality, feed-forward ANNs having one hidden layer of neurons are considered. These may be well represented by the expression,

$$\hat{y}(\mathbf{x}_k, \mathbf{w}) = \alpha_0 + \sum_{i=1}^n \alpha_i \varphi_i(\mathbf{x}_k, \beta_i), \quad (9)$$

where $\mathbf{w} = [\alpha, \beta]^T$ is the model parameter vector, $\alpha = [\alpha_0, \alpha_1, \dots, \alpha_n]$ is the vector of scalar linear parameters, and $\beta = [\beta_1, \dots, \beta_n]$ is composed of n β_i vectors of non-linear parameters, each one associated with one neuron. For a given set of input patterns \mathbf{X} , training the NN corresponds to finding the values of \mathbf{w} such that (10) is minimised:

$$\Omega(\mathbf{X}, \mathbf{w}) = \frac{1}{2} \|\mathbf{y} - \hat{\mathbf{y}}(\mathbf{X}, \mathbf{w})\|^2 \quad (10)$$

The $\frac{1}{2}$ factor is used for convenience considering the training algorithm to be employed. As the model output is a linear combination of the neuron activation functions output, (10) may be written as,

$$\Omega(\mathbf{X}, \mathbf{w}) = \frac{1}{2} \|\mathbf{y} - \phi(\mathbf{X}, \beta) \alpha\|^2 \quad (11)$$

where, omitting the dependence of ϕ on β ,

$$\phi(\mathbf{X}, \beta) = [\varphi(\mathbf{x}_1) \ \varphi(\mathbf{x}_2) \ \dots \ \varphi(\mathbf{x}_N)]^T.$$

By computing the optimal value α^* of the linear parameters α with respect to the non-linear parameters β , as a least-squares solution,

$$\alpha^* = \phi^+(\mathbf{X}, \beta) \mathbf{y}, \quad (12)$$

where "+" denotes a pseudo-inverse operation, and by replacing (12) in (11), the training criterion to compute the non-linear parameters is obtained:

$$\Psi(\mathbf{X}, \beta) = \frac{1}{2} \|\mathbf{y} - \phi(\mathbf{X}, \beta) \phi^+(\mathbf{X}, \beta) \mathbf{y}\|^2. \quad (13)$$

This criterion is independent of the linear parameters α and explicitly incorporates the finding that, whatever values the non-linear parameters β take, the α^* parameters employed are the optimal ones. Moreover, it reflects the non-linear/linear parameters structure of the feed-forward ANN model in (9), by separating their computation. This way it becomes possible to iteratively minimise (13) to find β^* , corresponding to searching for the best non-linear mapping, and then solve (12) using β^* to obtain the complete optimal parameter vector \mathbf{w}^* . The modified criterion enables the usage of appropriate methods to compute each type of parameters in the minimisation of a single explicit criterion. It lowers the dimensionality of the problem and usually achieves increased convergence rate.

2.4.2 Training Algorithm Various training algorithms can be employed to minimise (10) or (13). First-order gradient algorithms (known for MLPs as the *back-propagation* algorithm) or second-order methods, such as quasi-Newton, Gauss-Newton or LM can be employed as training algorithm. For non-linear least-squares problems the LM algorithm is recognised as the best method, as it exploits the sum-of-squares characteristic of the problem [38].

Denoting the standard (10) or modified (13) training criteria in iteration k by $\Omega(\mathbf{w}_k)$ (omitting the dependence of Ω on \mathbf{X}), a search direction \mathbf{p}_k in parameter space is computed such that $\Omega(\mathbf{w}_k + \mathbf{p}_k) < \Omega(\mathbf{w}_k)$. This method is said to be of the restricted step type because it attempts to define a neighbourhood of \mathbf{w}_k in which a quadratic function agrees with $\Omega(\mathbf{w}_k + \mathbf{p}_k)$ in some sense. The step \mathbf{p}_k is restricted by the region of validity of the quadratic function which is obtained by formulating in terms of \mathbf{p}_k a truncated Taylor series expansion of $\Omega(\mathbf{w}_k + \mathbf{p}_k)$. Then, it may be shown that \mathbf{p}_k can be obtained by solving the following system [31]:

$$(\mathbf{J}_k^T \mathbf{J}_k + \nu_k \mathbf{I}) \mathbf{p}_k = -\mathbf{g}_k. \quad (14)$$

\mathbf{g}_k and \mathbf{J}_k are, respectively, the gradient and Jacobean matrix of $\Omega(\mathbf{w}_k)$, $\nu_k \geq 0$ is a scalar controlling the magnitude and direction of \mathbf{p}_k . By recalling (3), the gradient may easily be obtained as,

$$\begin{aligned} \mathbf{g}_k &= \frac{\partial \Omega(\mathbf{w}_k)}{\partial \mathbf{w}_k} = \\ &= -\mathbf{J}_k^T \mathbf{e}_k, \end{aligned} \quad (15)$$

where the Jacobean matrix has the form:

$$\mathbf{J}_k = \begin{pmatrix} \frac{\partial y_1}{\partial w_1} & \dots & \frac{\partial y_1}{\partial w_l} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_N}{\partial w_1} & \dots & \frac{\partial y_N}{\partial w_l} \end{pmatrix} \quad (16)$$

The advantage of the LM algorithm is that in every iteration the value of ν is adapted in order to provide a step direction more close to the Gauss-Newton or gradient-descent methods. When $\nu \rightarrow 0$ the step direction approaches that of the Gauss-Newton method, when $\nu \rightarrow \infty$ it approaches the gradient-descent direction. Many variations of Marquardt's algorithm have been proposed concerning the rules governing the adaptation of ν . The original method [31] or a similar one [20, 11] should suit most applications.

2.4.3 RBF Network The RBF ANN is formulated by (9) where the i^{th} basis function or neuron, $\varphi_i(\mathbf{x}_k, \beta_i)$, is usually a Gaussian, a multiquadric, or an inverse multiquadric function. In most cases the Gaussian is employed:

$$\varphi_i(\mathbf{x}_k, \beta_i) = \exp\left(-\frac{1}{2\sigma_i^2} \|\mathbf{x}_k - \mathbf{c}_i\|^2\right) \quad (17)$$

In this case $\beta_i = [\mathbf{c}_i \ \sigma_i]$ is the non-linear parameter vector where \mathbf{c}_i is a point in input space, the centre of the Gaussian function, and σ_i is the corresponding spread. The outputs of all neurons are then linearly combined (recall Eq. 9) to produce the network output.

At the first iteration of the training algorithm the model parameters have to be initialised. Common approaches consist in selecting \mathbf{c}_i randomly from the input pat-

terns or from the input variables range of values. An alternative is to take advantage of clustering algorithms in order to spread the centres in distinct regions of the input feature space. The σ_i parameters may be chosen randomly, or, for instance using the simple rule [26, p. 299],

$$\sigma_i = \frac{z^{max}}{\sqrt{2n}}, \quad (18)$$

where z^{max} is the maximum Euclidean distance among the initial centres \mathbf{c}_i , and n is the number of neurons. Once the vectors of non-linear parameters, β_i , are initialized, (12) may be employed to determine the initial linear parameters, α_i .

In order to employ the LM algorithm to optimise the RBF ANN parameter vector, the error criterion must be defined as well as the derivatives required. If the standard formulation (10) is used, the three derivatives required to compute \mathbf{J} are:

$$\begin{aligned} \frac{\partial y}{\partial \mathbf{c}_i} &= \varphi_i(\mathbf{x}) \frac{\alpha_i}{\sigma_i^2} (\mathbf{x} - \mathbf{c}_i)^T, \\ \frac{\partial y}{\partial \sigma_i} &= \varphi_i(\mathbf{x}) \frac{\alpha_i}{\sigma_i^3} \|\mathbf{x} - \mathbf{c}_i\|^2, \\ \frac{\partial y}{\partial \alpha_i} &= \varphi_i(\mathbf{x}). \end{aligned} \quad (19)$$

For the modified criterion (13) alternative Jacobean matrices are available. It has been shown [39] that a simple and efficient solution consists in using the two first lines of (19), where the α_i are replaced by their optimal values as computed in (12). Remarkably, the use of this Jacobian matrix implies that each iteration of the LM method minimising (13) is computationally cheaper than minimising (10).

2.4.4 Stopping the Training Algorithm As most ANN training algorithms are iterative, some criteria is required to stop the training procedure after a certain number of iterations. Whichever method is employed, it should prevent the algorithm to *overtrain* the network parameters. Overtraining is a "phenomenon" likely to occur when using iterative training algorithms, characterised by a distinct behaviour of the error criterion when computed on two, distinct, data sets. On the data set employed to estimate the model parameters (training data set), the error criterion decreases with the number of iterations usually reaching a plateau where improvements become negligible. If a second data set is used to test the model at every iteration, then the error criterion taken on the testing data set decreases to a certain iteration and starts to increase in subsequent iterations. Beyond this point overtraining occurred because too many iterations were executed and the model became biased by the training examples, thus loosing the capability to generalise properly when presented with new input patterns. The methods of regularisation and *early stopping* are probably the most common to avoid overtraining. The first is a technique based on extending the error criterion with a penalty term, therefore numerically changing the training method, the second is a data driven cross-validation approach that may

be viewed as an implicit regularisation method. Interesting in-depth reading about the early stopping and overtraining subjects may be found in [1, 45].

In practice overtraining is avoided by stopping the training algorithm before reaching the absolute minimum of the training criterion. The method of early stopping, requires splitting the training input space \mathbf{F}^p (see Fig. 5, Sect. 2.3) into two data sets, the first, \mathbf{F}^t , to estimate the model parameters, called training data set, the second, \mathbf{F}^g , to assess the model generalisation capability, called generalisation data set. In the model design cycle presented earlier, by using a given MOEA individual chromosome, \mathbf{F} is indexed by the input terms in the chromosome. After indexing, the input data set is denoted by \mathbf{X} and the input-output data set by $\mathbf{D} = (\mathbf{X}, \mathbf{y})$. Consequently the training, generalisation, and validation data sets, for a given ANN, will be defined as,

$$\begin{aligned}\mathbf{D}^t &= (\mathbf{X}^t, \mathbf{y}^t) , \\ \mathbf{D}^g &= (\mathbf{X}^g, \mathbf{y}^g) , \\ \mathbf{D}^v &= (\mathbf{X}^v, \mathbf{y}^v) ,\end{aligned}$$

for training, generalisation testing, and validation, respectively. Recall that \mathbf{D}^v is meant to validate the MOEA model optimisation globally, to avoid bias towards \mathbf{D}^t and \mathbf{D}^g in the final model selection. The proportions of points from \mathbf{D} that compose \mathbf{D}^t and \mathbf{D}^g are often selected in an ad hoc fashion, usually by means of trial and error. A statistically validated principled way of selecting that proportion may be found in [1].

By denoting the error criterion computed on the testing and generalisation data sets at iteration k by $\Omega(\mathbf{D}^t, \mathbf{w}_k^*)$ and $\Omega(\mathbf{D}^g, \mathbf{w}_k^*)$, the early stopping method consists in selecting the model parameters corresponding to the iteration where $\Omega(\mathbf{D}^g, \mathbf{w}_k^*)$ ceased to decrease (assuming $\Omega(\mathbf{D}^t, \mathbf{w}_k^*) \leq \Omega(\mathbf{D}^t, \mathbf{w}_{k-1}^*)$). In practice the inflection point on the $\Omega(\mathbf{D}^g, \mathbf{w}_k^*)$ curve must not be identified locally by a rule of the type $\Omega(\mathbf{D}^g, \mathbf{w}_k^*) > \Omega(\mathbf{D}^g, \mathbf{w}_{k-1}^*)$ as this method would be sensitive to small variations that are still occurring in a more global descending trend. An alternative is to define k^{max} as the maximum number of iterations to execute and then find the global minimum of $\Omega(\mathbf{D}^g, \mathbf{w}_k^*)$. Formally, assuming monotonically decreasing $\Omega(\mathbf{D}^t, \mathbf{w}_k^*)$, this may be written as,

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \{ \Omega(\mathbf{D}^g, \mathbf{w}_k^*) \}_{k=1}^{k^{max}} , \quad (20)$$

where k^{max} should be large enough to include the global minimum of $\Omega(\mathbf{D}^g, \mathbf{w}_k^*)$. When this method is used to stop the training algorithm used in a MOEA model identification experiment, $\Omega(\mathbf{D}^t, \mathbf{w}^*)$ and $\Omega(\mathbf{D}^g, \mathbf{w}^*)$ are commonly included in μ^p expressing the goal of identifying models achieving a good data fitting and good generalisation capability.

If early-stopping is not possible, an alternative is to employ a set of termination criteria that is commonly used in unconstrained optimisation [23, p. 306]. Let θ_k , a measure of absolute accuracy, be defined as,

$$\theta_k = \tau^\Omega \times (1 + |\Omega(\mathbf{D}^t, \mathbf{w}_k^*)|) , \quad (21)$$

where τ^Ω is a measure of the desired number of correct digits in the objective function. The optimisation stops when all the following conditions are met:

$$\Omega(\mathbf{D}^l, \mathbf{w}_{k-1}^*) - \Omega(\mathbf{D}^l, \mathbf{w}_k^*) < \theta_k \quad (22)$$

$$\|\mathbf{w}_{k-1}^* - \mathbf{w}_k^*\| < \sqrt{\tau^\Omega} (1 + \|\mathbf{w}_k^*\|) \quad (23)$$

$$\|\mathbf{g}_k\| \leq \sqrt[3]{\tau^\Omega} (1 + |\Omega(\mathbf{D}^l, \mathbf{w}_k^*)|) \quad (24)$$

The two first conditions test the convergence of the model parameters. The reasoning behind the use of two conditions is that for ill-conditioned problems, $\Omega(\mathbf{D}^l, \mathbf{w}_k^*)$ may be a good approximation of the global minimum (22), but \mathbf{w}_k^* may be far from the optimum and the algorithm may still be making large adjustments to \mathbf{w}^* (23). The third condition reflects the necessity that the gradient should be near zero if $\Omega(\mathbf{D}^l, \mathbf{w}_k^*)$ is close to the optimum. This method achieves a certain level of regularisation, implicitly related to the parameter τ^Ω , and does not require the \mathbf{D}^g data set, therefore lowering the number of function evaluations required by the inclusion of $\Omega(\mathbf{D}^g, \mathbf{w}_k^*)$ in the objective space of the model identification problem. The disadvantage is that the resulting number of training iterations might not be enough to adequately converge the model parameters, or it might be in excess and provoke overtraining.

3 Example Model Identification Problems

To exemplify the use of the methodologies presented in the sections above, two ANN model identification problems that the authors have been involved with will be discussed. The first deals with the prediction of the Portuguese electricity consumption profile within an horizon of 48 hours, the second is related to the estimation of cloudiness from ground-based all-sky hemispherical digital images.

3.1 Electricity Consumption Prediction

The Portuguese power grid company, *Rede Eléctrica Nacional* (REN), aims to employ electricity load demand (ELD) predictive models on-line in their dispatch system to identify the need of reserves to be allocated in the Iberian market. To accomplish this the evolution of ELD over a prediction horizon of at least 48 hours is required. The problem is addressed from the point of view of identifying RBF ANN one-step-ahead ELD predictive models using the framework already described. These models are iterated in a multi-step fashion in order to predict the electricity consumption profile up to the specified prediction horizon.

In previous work [19] a literature review on the ELD forecasting area was presented, demonstrating that the approach taken was relevant and ambitious as no publications were found considering simultaneously four aspects that the team is actively addressing:

Prediction scheme	To meet the requirement, one-step-ahead predictive models are iterated in a multi-step fashion in order to obtain the consumption profile up to the specified prediction horizon. Most work relies on the prediction of daily peak consumption or accumulated consumption over a certain period, and does not consider dynamics.
Model adaptation	On-line model adaptation strategies are necessary, as the models are static mappings with external dynamics and the profiles of electricity consumption vary over time.
Perturbations	One input is incorporated in the models to account for the effect of events that dramatically perturb the typical profile of load demand (the effect of week-ends, holidays and other foreseeable events).
Optimised models	The problem of model structure optimisation and selection is clearly formulated and approached by appropriate methodologies in order to meet specified design requirements.

By that time a number of exploratory identification experiments were executed which allowed refining the problem formulation (see Sect. 2.3, model design cycle). In the following, one last identification experiment is described, from which one model was selected [18]. It is currently in operation at the Portuguese power-grid company dispatch system.

3.1.1 Problem Formulation Two types of model structures were previously compared [19], the NAR and the NARX. For the latter only one exogenous input was considered, encoding the occurrence of events perturbing the daily and weekly patterns of electricity consumption. That comparison favoured the NARX approach as it consistently achieved a considerably better prediction accuracy.

Table 1 Day of the week and holiday occurrence encoding values

Day of week	Regular day	Holiday	Special
Monday	0.05	0.40	0.70
Tuesday	0.10	0.80	
Wednesday	0.15	0.50	
Thursday	0.20	1.00	
Friday	0.25	0.60	0.90
Saturday	0.30	0.30	
Sunday	0.35	0.35	

The exogenous input encoding, presented in table 1, distinguishes between the days of the week and also the occurrence and severity of holidays based on the day of their occurrence. The *regular day* column shows the coding for the days that are not

holidays. The next column presents the encoded values when there is a holiday for that day of the week, and finally, the *special* column shows the values that substitute the regular day value in two special cases: for Mondays when Tuesday is a holiday; and, for Fridays when Thursday is a holiday. Figure 6 illustrates the severity of the perturbation that a holiday causes in the electricity consumption profile. It is evident

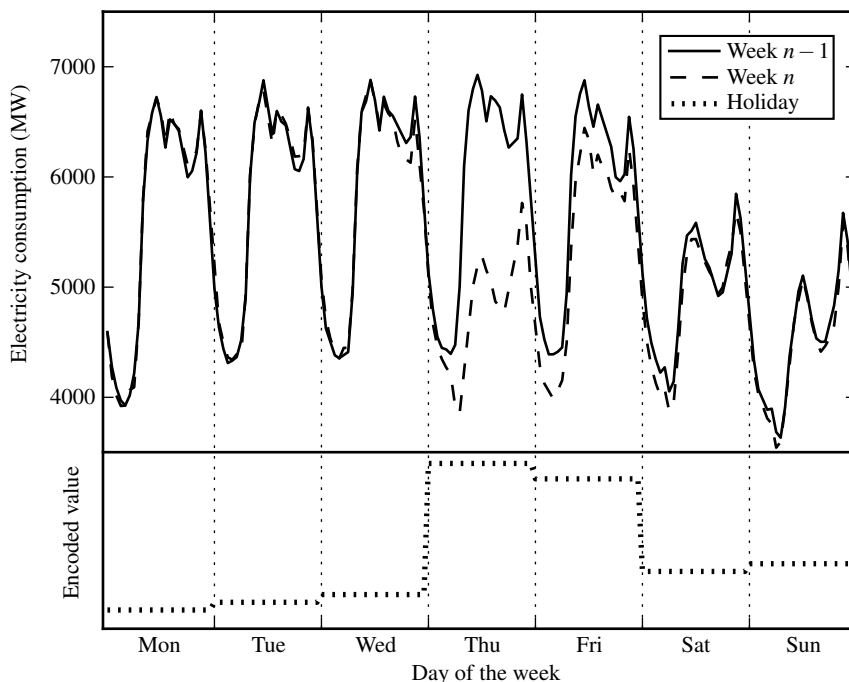


Fig. 6 ELD in two consecutive weeks. A holiday occurs in the second week.

that not only during the holiday the change is very large, but also on the following day a significant change may be observed.

The data used in the model identification experiment corresponds to the Portuguese electrical energy consumption measured at hourly intervals, for a time span starting around mid October 2007, and ranging to the end of 2008. The complete time-series is presented in Fig. 7. It was split in four data sets for model training, generalisation testing, predictive simulation, and validation. The points for each set were selected from three distinct periods of the year, delimited in Fig. 7 by two vertical dotted lines to the right of the plot. \mathbf{D}^t and \mathbf{D}^s are composed of 330 and 60 days of data points randomly selected from the first period. The last 50 days of 2008 were divided in two parts, the first being used as a simulation data set, \mathbf{D}^s , the second as the validation data set, \mathbf{D}^v . Taking into account the use of one input encoding the

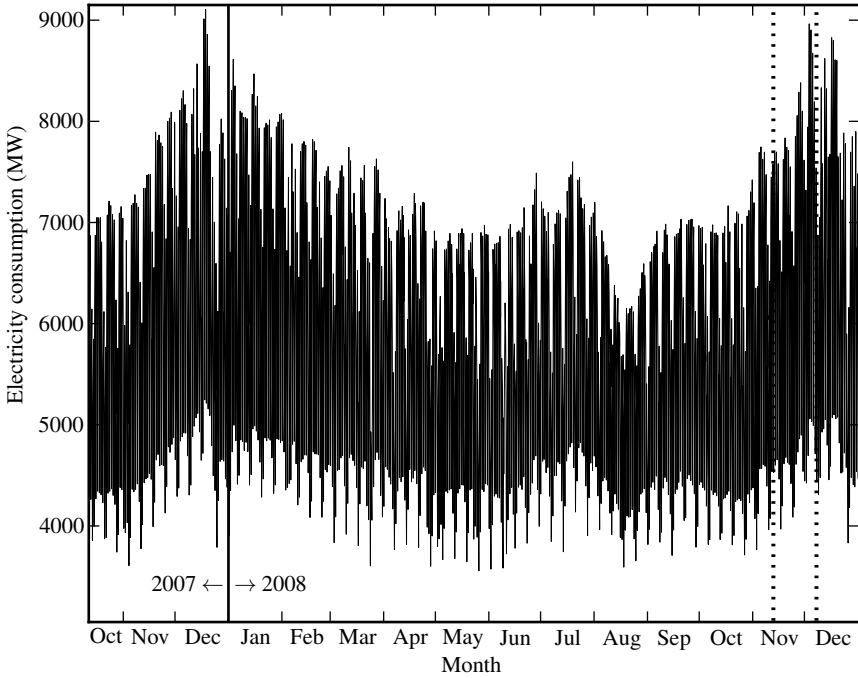


Fig. 7 ELD in Portugal for the time span considered

occurrence of weekends and holidays, care was taken to make sure that the data sets \mathbf{D}^s and \mathbf{D}^v included holidays.

The lookup table \mathbf{F} , from which the four data sets are built for each ANN by indexing using the input part of the chromosomes, is composed of 168 delayed ELD input terms plus the input encoding the occurrence and severity of holidays. The ELD input terms correspond to one week window, an interval for which the time-series exhibits a clear repetitive pattern. This pool of candidate input terms was specified by considering the results of previous experiments. Also the limits for the number of neurons and for the size of the input part of the chromosome were specified by taking previous results into account. In this case significant changes were made by doubling the maximum number of neurons, $n \in [10, 28]$, and by increasing the maximum number of input terms allowed, $d \in [2, 40]$.

The model parameters were estimated via the LM algorithm using the modified training criterion (13) as outlined in Sect. 2.4. The initial centre locations for the Gaussian activation functions were selected randomly from the input patterns in \mathbf{D}^f , the corresponding initial spreads, σ_i , were determined by the rule (18), and the linear parameters were initialised using (12). The early stopping method was employed to stop the training algorithm by setting $k^{max} = 200$ in (20).

In order to address the model structure selection problem, the multiobjective genetic algorithm (MOGA) [22] is employed to evolve a *preferable set* of models whose number of neurons and selected input terms optimise a number of pre-

specified goals and objectives. These, as discussed in Sect. 2.1, are specified by a two component vector of objective functions, $\mu = [\mu^p, \mu^s]$. For the first component, related to the ANNs parameter training process, two model performance objectives were considered, given by the root-mean-square (RMS) error computed on the training and generalisation testing data sets, respectively denoted by $\rho(\mathbf{D}^t)$ and $\rho(\mathbf{D}^s)$. The first one is used as a restriction because a clear positive (linear or not) relationship between the training criterion and a long-term prediction performance (to be defined below) is not guaranteed and was not observed in practice in previous experiments. One last objective was specified for the first component of μ , given by the 2-norm of the linear parameters vector, $\|\alpha\|$. It is employed as a restriction in order to guarantee good numerical properties and parameter convergence in the models, but in fact it also acts as a penalty term for the complexity of the model. Regarding μ^s , the component of μ related to the model structure selection and to the specific model application, one objective was considered expressing the final goal of the model application: the prediction of the electricity consumption profile within an horizon of 48 hours. It is computed on the basis of the long-term model prediction error taken from the multi-step model simulation over the prediction horizon ph . Assume that a given simulation data set, \mathbf{D} , has p data points and for each point the model is used to make predictions up to ph steps ahead. Then an error matrix is constructed,

$$\mathbf{E}(\mathbf{D}, ph) = \begin{pmatrix} e[1, 1] & e[1, 2] & \cdots & e[1, ph] \\ e[2, 1] & e[2, 2] & \cdots & e[2, ph] \\ \vdots & \vdots & \ddots & \vdots \\ e[p - ph, 1] & e[p - ph, 2] & \cdots & e[p - ph, ph] \end{pmatrix},$$

where $e[i, j]$ is the model prediction error taken from instant i of \mathbf{D} , at step j within the prediction horizon. Denoting the RMS function operating over the i^{th} column of its argument matrix by $\rho(\cdot, i)$, then the long term prediction performance measure is defined as,

$$\varepsilon(\mathbf{D}, ph) = \sum_{i=1}^{ph} \rho(\mathbf{E}(\mathbf{D}, ph), i), \quad (25)$$

which is simply the summed RMS of the columns of \mathbf{E} . This way the single objective in μ^p is simply given by $\varepsilon(\mathbf{D}^s, 48)$. This represented a considerable change from previous work as the prediction horizon was doubled from 24 to 48 hours.

The complete objective vector for the ELD prediction problem is therefore specified as $\mu = [\rho(\mathbf{D}^t) \quad \rho(\mathbf{D}^s) \quad \|\alpha\| \quad \varepsilon(\mathbf{D}^s, 48)]$. Table 2 summarises the objectives and their configuration as used in the MOGA ELD predictive modelling experiment. As the ANN parameters are randomly initialised, for each individual, 10 training trials were executed and the averages of $\rho(\mathbf{D}^t)$ and $\rho(\mathbf{D}^s)$ were used for evaluation purposes. This procedure decreases the likelihood of unrealistic fitness assignment in the MOEA as one good ANN structure could be poorly evaluated due to a bad choice of initial parameters. In order to decrease the computational load, $\varepsilon(\mathbf{D}^s, 48)$

Table 2 Objective space configuration for the MOGA ELD prediction problem

μ component	Objective function	Set up as
μ^p	$\rho(\mathbf{D}^f)$	restriction < 100 MW
	$\rho(\mathbf{D}^g)$	to minimise
	$\ \alpha\ $	restriction < 200
μ^s	$\varepsilon(\mathbf{D}^s, 48)$	minimise

was only computed for the trial instance whose pair $\{\rho(\mathbf{D}^f), \rho(\mathbf{D}^g)\}$ is closer (in the Euclidean sense) to the averages over the 10 trials.

3.1.2 Results and Discussion Figure 8 illustrates the results obtained in the space of objectives after 50 generations of the MOGA (values in Mega Watt (MW)). At this generation the execution

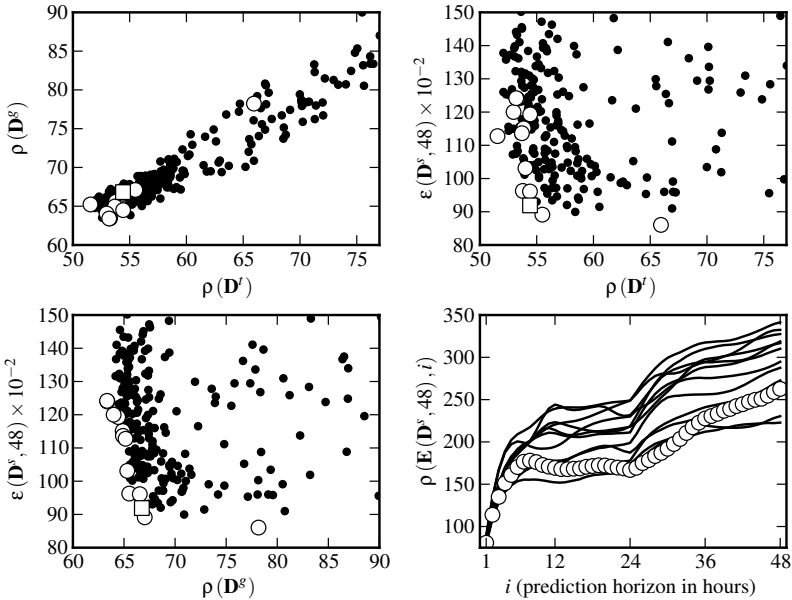
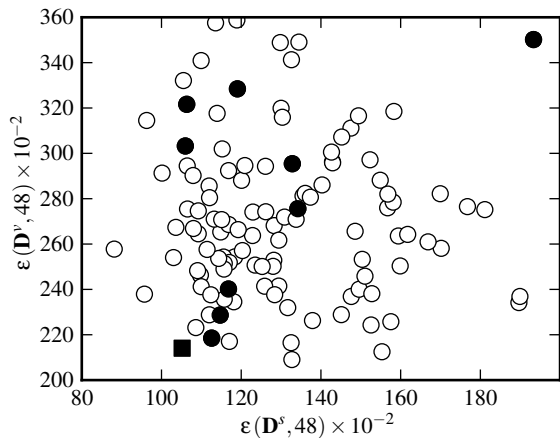


Fig. 8 Objectives of the MOGA ELD predictive model identification experiment

was stopped as the last models having lower values on $\varepsilon(\mathbf{D}^s, 48)$ entered the preferable set on generation 30, and convergence was only occurring for $\rho(\mathbf{D}^t)$ and $\rho(\mathbf{D}^g)$ with no effect on $\varepsilon(\mathbf{D}^s, 48)$. This considerably smaller number of generations, when compared to previous work, clearly shows the benefits of using averaged objective values over multiple model training trials. The three scatter plots show the results of non-dominated individuals using dark points, and the results of the 13 models in the preferable set using white circles (and one white square). The top-left plot shows a linear relation between the error criterion obtained on the training and generalisation testing data sets. The plots at the top-right and bottom-left show the relation between each error criterion and the long-term prediction error measure, where the conflict between these objectives is well demonstrated. The lower-right plot presents the evolution of $\rho(\mathbf{E}(\mathbf{D}^s, ph), i)$ with i from 1 to ph , the prediction horizon. The curve marked with white circles was obtained by the model marked using a white square on the remaining plots. The objective values are slightly better than those obtained in previous work, however it should be noted that the prediction horizon was doubled. The 13 selected models had from 24 to 28 neurons, 26 the most frequent, and from 34 to 39 input terms, 36 the most frequent. All of them included the holiday encoding input, and other 15 input terms were employed in 10 models or more. When compared to previous work, the increase in model complexity is explained by a slightly better predictive accuracy over a double size prediction horizon.

The models obtained were evaluated on the validation data set, \mathbf{D}^v , in order to select one for further assessment of predictive accuracy and robustness. Considering that during the MOGA execution only one out of 10 models was evaluated for $\varepsilon(\mathbf{D}^s, 48)$, for each of the models in the preferable set 10 further training trials were executed and their performance was evaluated on \mathbf{D}^s and \mathbf{D}^v . The results are depicted in Fig. 9 where the dark markers highlight the results obtained by the chosen model structure (marked by white squares in Fig. 8), the square marker corresponding to the instance selected for further study. This was accomplished by comparing

Fig. 9 Detail of the long term prediction error measure obtained on the simulation data set, \mathbf{D}^s , versus that obtained in the validation data set, \mathbf{D}^v , for the preferable set of 13 models. Ten training trials per model.



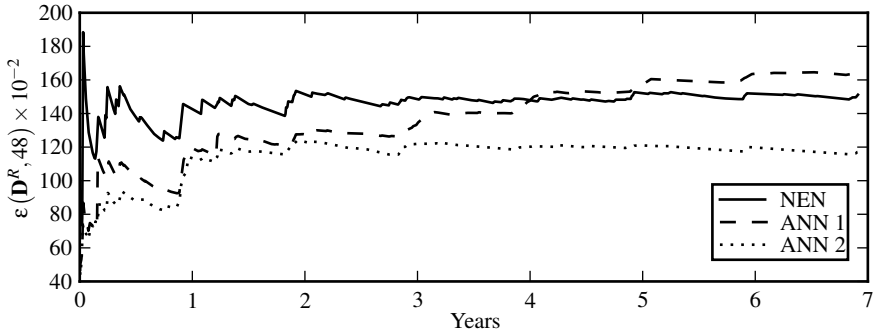


Fig. 10 Comparison between the selected ANN (ANN 1), a nearest neighbour approach (NEN), and the selected ANN with a yearly retraining (ANN 2)

the selected model structure to a nearest neighbour (NEN) predictive approach on a data set (denoted by \mathbf{D}^R) ranging from the beginning of 2001 to the end of 2008. A complete description of the NEN methodology is given in [18]. It was tested by varying the number of nearest neighbours employed for prediction and by using a sliding window of past 54 weeks to conduct the nearest neighbour search. The selected ANN was used to predict the ELD on the same instants (≈ 7 years) as the NEN method, being initially trained using data from the first sliding window (approximately the 2001 year). Figure 10 shows the evolution of $\epsilon(\mathbf{D}^R, 48)$ as time increases. It may be seen that the NEN method is quite robust as the prediction performance measure converges asymptotically to a value near 150×10^{-2} . The line labelled ANN 1, corresponding to the model structure selected, shows that as time passes $\epsilon(\mathbf{D}^R, 48)$ tends to increase at an almost constant rate, becoming higher than that of the NEN method after about 4 years of data. This is likely to happen because the ANN parameters no longer reflect with the same accuracy the underlying dynamics and trend of the ELD time series, leading to the conclusion that the ANN requires some form of adaptation to become robust. Even so it is quite remarkable that, with no parameter change, it achieves better prediction accuracy during the first four years of data (significantly better in the first three years). In order to obtain a fair comparison with the NEN method (in the sense that it uses a sliding window of information), another set of results was computed by retraining the ANN at every year interval so that its parameters are readjusted to reflect more closely the ELD data dynamics and trends. These results are labelled ANN 2 in Fig. 10, showing that the improvements are significant even though only a yearly retraining was employed. In terms of robustness this is very promising for the actual implementation of ANN ELD predictive models in the REN dispatch system, as further improvements are expectable if more elaborate model adaptation techniques are employed or a more frequent retraining is used [15]. Figure 11 shows the evolution of the ELD over the prediction horizon for the simulation instant where the retrained ANN achieved the RMS error value which was closest to the average over the complete simulation. The

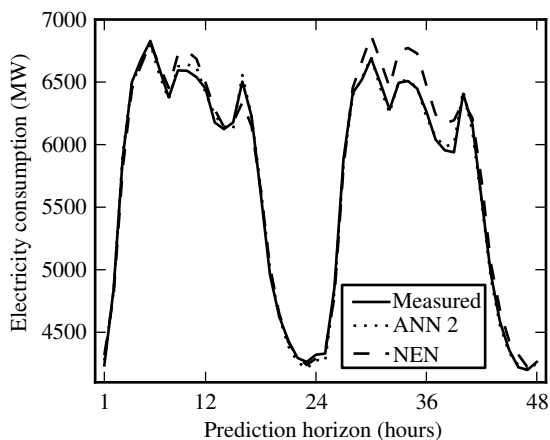


Fig. 11 Prediction horizon where the retrained model (ANN 2) RMS of error is closest to the average obtained in the complete simulation. The nearest neighbour approach prediction is shown for comparison.

prediction obtained by the NEN method is also shown for comparison. Globally the results show that the ANN is preferable to the NEN method, although at the cost of a significant increase in methodology complexity and on computational effort.

In summary, when compared to previous results, better prediction accuracy was achieved over a longer prediction horizon, and faster convergence (in number of generations) was observed in the MOGA execution.

3.2 Cloudiness Estimation

Clouds are an important phenomena strongly affecting the total incoming irradiance at a given point in the Earth surface. For a growing number of applications in diverse fields such as agriculture, forestry or energy production and management, being able to accurately estimate and predict solar radiation at a given ground location and at short time scales, is becoming an extremely important task because solar radiation strongly influences the relevant processes and energy balances. The use of ground-based all-sky (GBAS) images acquired by CCD cameras, directly with fish-eye lenses or projected on hemispherical mirrors, has been receiving growing interest by researchers from several fields (See [12] for examples and references). Regarding the use of cloudiness information extracted from GBAS images and its incorporation into solar radiation predictive modelling, our group made a first attempt in a previous work [9]. The pixel classification approach was quite different from that being presented here and there was no assessment, other than by visual inspection, on the accuracy of the cloud cover estimation. By that time no clear conclusion could be made on the benefits of using cloudiness information at the inputs of the neural network predictive solar radiation model. Typically predictive solar radiation models are identified using one-step-ahead NAR forms. Due to the autoregressive characteristic their accuracy becomes severely degraded in the presence of cloudy sky conditions, hence the need to advance to the NARX form, considering cloudiness has the exogenous input.

The motivation for this work is twofold: to improve the predictive performance of global solar radiation models operating on relatively short time scales (prediction horizons of a few hours); and, to implement these models on a cheap hardware daytime all-sky imaging prototype being developed in the laboratory. Ultimately our goals are to employ global solar radiation predictive models incorporating the effects of cloudiness in projects related to the efficient energy management in public buildings and, in the future, in projects related to solar power plants and to the prediction of electricity consumption.

3.2.1 Problem Formulation A total of 410 all-sky images were used in the model identification experiment. They were acquired using a *Total Sky Imager (TSI) 440A* manufactured by *Yankee Environmental Systems, Inc.*, located on top of one building ($37^{\circ}02'N$, $07^{\circ}57'W$) in the University of Algarve, Faro, Portugal. The images are stored in red-green-blue (RGB) colour mode (8 bit/channel) with a dimension of 704×576 (width \times height). Given the location of the TSI and the time-stamp of each image, a pixel mask was computed to identify the visible sky pixels for further processing (see Fig. 13 for an example). For these, one researcher made an additional mask including all the cloud pixels according to his personal judgement. Using these masks the percent cloud cover for each image was computed using the formula,

$$C = \frac{N_c}{N_s + N_c} \times 100, \quad (26)$$

where N_s and N_c are the numbers of pixels masked as clear sky (class **S**) and cloud (class **C**), respectively. Figure 12 presents information about the images used, illustrating the effort made to include significant numbers of images within intervals of the cloud cover and the time of day. Additionally, for each pixel intensity scale considered and for every image, an exhaustive search was conducted to find the threshold value, t_o , minimising the cloud cover estimation error resulting from the thresholding operation.

The general approach consists in finding a threshold value, \hat{t} , on a given pixel intensity scale, which segments the image I pixels with coordinates (x, y) and intensity γ_{xy} into one of the classes, **S** and **C**. In this sense these are sets defined as,

$$\begin{aligned} \mathbf{S} &= \{(x, y) \in I : \gamma_{xy} \leq \hat{t}\}, \\ \mathbf{C} &= \{(x, y) \in I : \gamma_{xy} > \hat{t}\}, \end{aligned}$$

to which N_s and N_c in (26) are the respective set cardinalities. The evaluation of thresholding methods relies on the absolute error between the cloud fraction attributed to the images and that estimated by the threshold \hat{t} :

$$\varepsilon = |C - \hat{C}_{\hat{t}}|. \quad (27)$$

Several pixel intensity scales were considered to perform the thresholding operation. From the results in [12] one, denoted *hsvR*, was selected as it consistently provided increased cloud cover estimation accuracy for various thresholding methods tested.

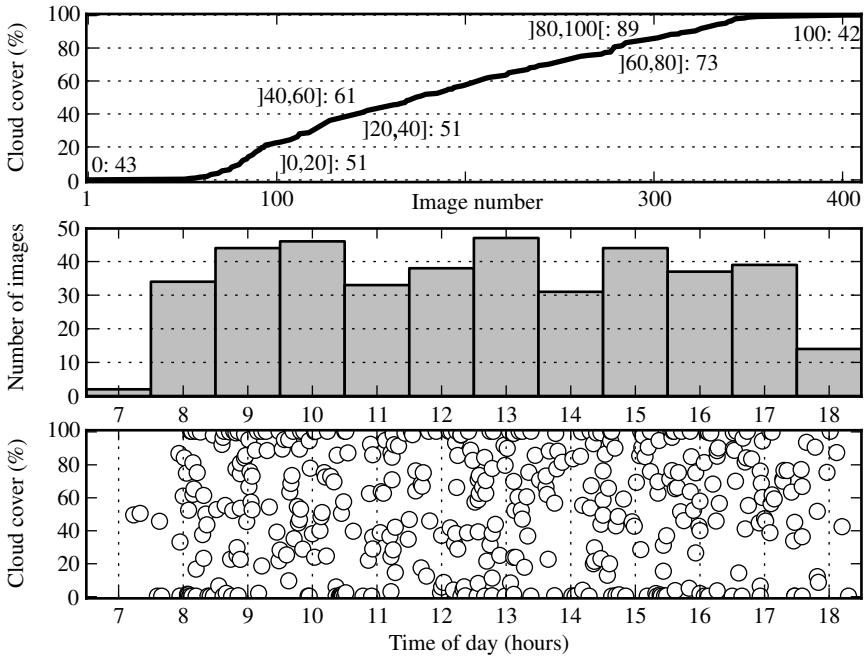


Fig. 12 Details about the 410 images used. Top: Number of images by ascending cloud cover intervals. Middle and bottom: respectively, the number of images and cloud cover distribution by the time of day.

This pixel intensity is obtained by converting the original RGB image to the *hue-saturation-value* (HSV) colour model, setting the value channel to 1 (the maximum) for all pixels, and finally converting this image back to the RGB mode. Setting an equal value on the V channel has an equalisation effect on the pixels luminosity. The maximum value was chosen because on the HSV colour model the colours become more distinguishable. The net effect on the converted RGB image is that clear sky and cloud pixels have improved contrast between them in the red channel.

The model identification problem consists in using the framework presented in Sect. 2 in order to search for a RBF ANN image segmentation model. As illustrated in Fig. 13, the output of the ANN is the threshold to be used in an image, the inputs are a set of features extracted from the masked image or transformations of it.

The set of 410 images was broken into three sub-sets: the training set, denoted by \mathbf{D}^t (290 images); the testing set, \mathbf{D}^g (60 images), for generalisation testing; and the validation set, \mathbf{D}^v (60 images), to evaluate the ANNs after the MOEA execution. From all the images and from transformations of them, a total of 69 features were extracted from distinct pixel intensity scales: first, from the original RGB image the HSV and hue-saturation-lightness (HSL) images were obtained; secondly, on the HSV and HSL images, the V and L channels were set to 1 and 0.5, respectively, and these transformed images were converted back to RGB mode, thus generating

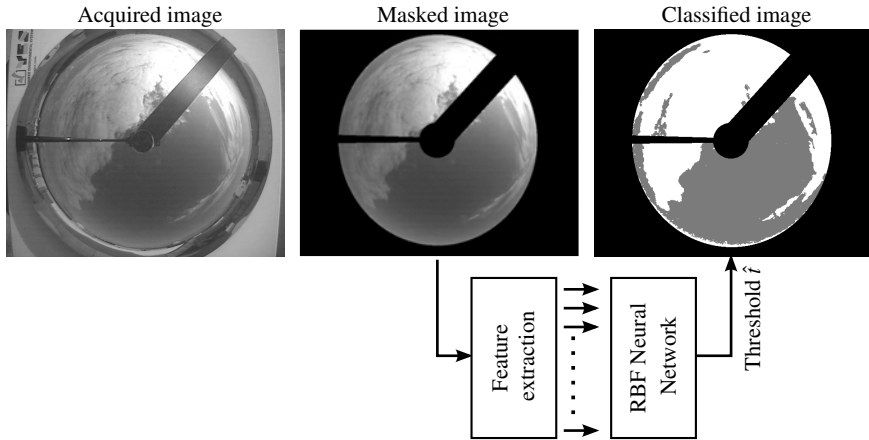


Fig. 13 Neural network image segmentation approach.

two additional RGB images; finally, from each RGB mode image, a grey intensity image was generated. This results in a total of 7 different images and 17 distinct intensity channels. From the latter, the sample mean, standard deviation, and skewness were extracted. Additionally, from the red and grey intensity channels (6 in total) histogram, the most frequent, first non-zero, and last non-zero intensity levels were also extracted.

From the lookup table, **F**, of 69 features, the model chromosomes were allowed to have $d \in [2, 36]$ input terms. The number of neurons, n , was restricted to the interval $[2, 24]$. As in the electricity consumption prediction problem, the model parameters were estimated via the LM algorithm using the modified training criterion (13) as outlined in Sect. 2.4. The initial centre locations for the Gaussian activation functions were selected randomly from the input patterns in \mathbf{D}^f , the corresponding initial spreads, σ_i , were determined by the rule (18), and the linear parameters were initialised using (12). The early stopping method was employed to stop the training algorithm by setting $k^{max} = 50$ in (20).

The MOGA was also employed to evolve a set of models whose selected number of neurons and input terms optimise a number of pre-specified goals and objectives. To this respect two objectives were set-up for minimisation: the RMS of the error computed on \mathbf{D}^f and on \mathbf{D}^g , respectively denoted by $\rho(\mathbf{D}^f)$ and $\rho(\mathbf{D}^g)$. As the ANNs are randomly initialised, for each of them 25 training trials were executed and the average of both objectives was used for evaluation purposes. Recall that this procedure decreases the chance of one potentially good model being poorly evaluated due to a bad choice of initial parameters. Once the MOGA execution was terminated, for each of the preferable ANN models a larger number of training trials was executed in order to select one model for application. This choice was made by taking into account the actual objective values attained on each of the trials and also the RMS output error obtained on the validation data set.

The model that was selected from the identification experiment was compared to other thresholding methodologies [12], namely, a fix threshold approach, the Ridler, Calvard and Trussel (RCT) algorithm [35, 46], and Otsu's method [33]. For the first, an histogram based analysis was made in order to identify the best single threshold value that could be applied to all the 410 images. Global (over all the images) pixel intensity probability mass functions were separately computed for each of the classes, **S** and **C**. Then, a search was conducted in a vicinity around the intersection point of the PMFs in order to find the threshold minimising the average (over all images) value of (27). This was found to be $t = 158$ on the $hsvR$ pixel intensity scale (the same used for the ANN approach).

The RCT method is an histogram-iteration form [46] of an iterative thresholding algorithm [35] that we denote by RCT in the following. A brief description of the method may be found in [12]. For a more in-depth view the reader should consult [35, 46, 30, 44]. Briefly, the RCT method tries to iteratively estimate the average pixel intensity of both classes and computes the threshold as the average of the classes sample mean.

The principle behind the method proposed by [33] is very simple: an exhaustive search is conducted on the pixel intensity scale for the threshold that maximises the inter-class variance. Again a brief overview may be found in [12], whereas for more detailed descriptions [33] or [44] may be consulted.

3.2.2 Results and Discussion The MOGA execution was stopped after 50 generations yielding 11 ANNs in the Pareto front as highlighted in the top-left plot of Fig. 14, where a detail of the objective values is shown. Regarding the number of neurons of the 11 selected models, four of them had from 12 to 14, the remaining seven had 22 or 23 neurons. Concerning the number of input features, the models employed from 29 to 36.

As mentioned before, 50 additional training trials were executed for each model selected. The resulting objective values are depicted in the top-right plot of Fig. 14. The plots at the bottom of the figure show the corresponding results considering the evaluation of each model structure instance on the validation data set: the RMS error obtained on \mathbf{D}^f and \mathbf{D}^g is plotted against the RMS error obtained in the validation data set, \mathbf{D}^v . The results marked with a dark square were obtained by the RBF ANN that was selected after analysis of all the results. It presented the most favourable balance in the objectives, achieving the RMS error values of 13.10, 13.12, and 14.65, respectively on \mathbf{D}^f , \mathbf{D}^g , and \mathbf{D}^v . It is a network with 30 inputs and 22 neurons.

Regarding the cloud cover fraction estimation, Table 3 presents the minimum, average, and maximum ε results obtained by the ANN model selected and by the remaining methods employed for comparison. For the ANN, they are presented considering the training and testing data sets together (involved in the MOGA ANN optimisation), the validation data set alone, and the three data sets altogether. It may be seen that the RCT and Otsu methods achieve similar results to those obtained with a fixed threshold. The results obtained by the RBF ANN selected using the framework presented in Sect. 2 represent an improvement in average accuracy of

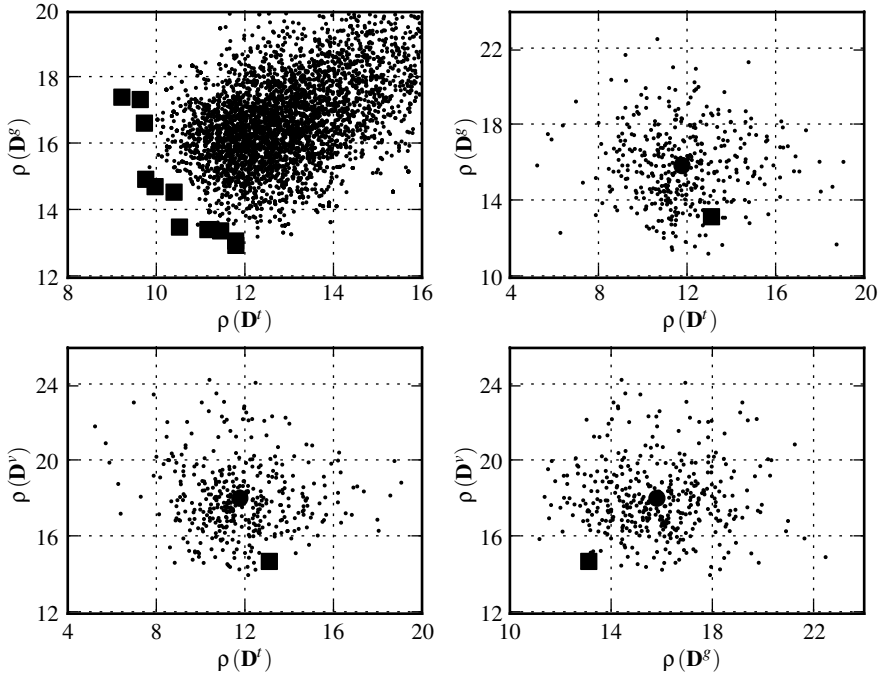


Fig. 14 MOGA results

Table 3 Absolute error of cloud cover estimation obtained by the RFB ANN image threshold approach (first three lines). Results obtained by three additional methods are shown for comparison (last three lines).

		minimum	average	maximum
Data set:	Training and testing	0.00	5.31	58.46
	Validation	0.00	4.74	43.71
	Altogether	0.00	5.22	58.46
Other methods:	Fixed threshold	0.00	11.24	82.64
	RCT method	0.00	11.34	98.21
	Otsu's method	0.00	11.07	63.59

approximately 50% when compared to the best results obtained by the remaining methods.

Figure 15 presents the absolute error values for the reference cloudiness of each image, where, for the ANN plot (bottom), the circles correspond to images in the training or testing data sets, and the dark squares to images in the validation data set. The similarity of results achieved by the three methods used for comparison is

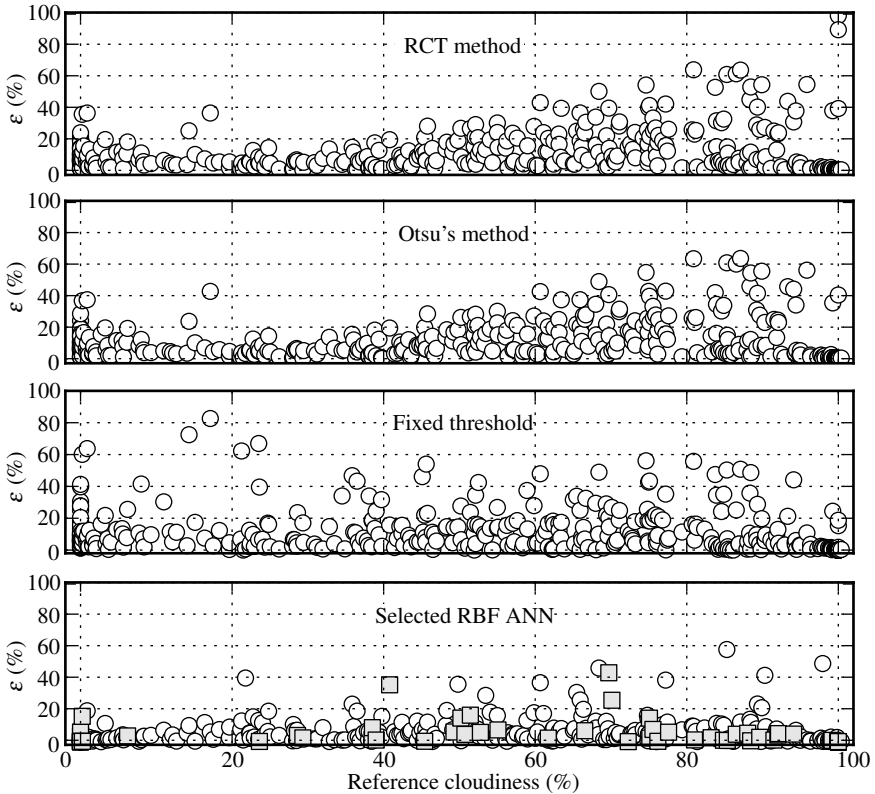


Fig. 15 Error performance of the three methods used for comparison, and of the RBF ANN image thresholding approach (bottom)

visible, although the fixed threshold approach exhibits improved uniformity of the error for the reference cloudiness when compared to the RCT and Otsu's methods. The improvement achieved by the selected ANN model is noticeable with most error values under 20%.

Despite the improvement obtained by the RBF ANN thresholding methodology there are a few directions in future work expected to further improve the results. Perhaps the most important, regarding the use of the MOEA to select ANNs, consists in specifying the objective space in a different way. In most images ϵ is not symmetric around the optimum threshold, thus minimising the threshold estimation error may not guarantee the best results. A better approach would consist in building a matrix where for each image (lines) the value of ϵ is computed for each pixel

intensity (columns), so that it becomes possible to map the NN threshold estimation to a cloud cover estimation error. The latter should be minimised in the MOGA search for NN structures. This is currently being implemented and will result in another iteration of the model design cycle. Once this is carried out, the resulting RBF ANN will be used to build a time-series of cloudiness from an existing time-series of GBAS images. Then a cloudiness predictive model will be identified and employed for the benefit of global solar radiation predictive models identification having cloudiness as an exogenous input. The goal is to conclude if that approach is preferable to auto-regressive solar radiation predictive models.

4 Concluding Remarks

Neural network modelling is an iterative process, requiring, at the present stage, substantial skills and knowledge from the designer. It is our view that the methodology presented in this article, employing multiobjective evolutionary algorithms for the design of neural models, is a suitable tool to aid the designer in this task. It incorporates inputs, model order and structure selection, as well as parameter estimation, providing the designer with a good number of well performing models with varying degrees of complexity. Importantly, the model identification framework is suitable, with minor adaptation, to most feed-forward artificial neural network methodologies. It also allows the incorporation of objectives which are specifically designed by considering the final application of the model. Through the analysis of the results obtained in one iteration, the search space can be reduced for future iterations, therefore allowing a more refined search in promising model regions. This was demonstrated in practice, by the presentation of two model identification experiments that were designed by taking into account results from previously executed experiments. In both, significant improvements were achieved not only when compared to previous work, but also by comparison with different methodologies.

Acknowledgements. The authors thank the Portuguese National Science Foundation for funding this work with project FCT PTDC/ENR/73345/2006, the Portuguese Power Grid company, "REN - Rede Eléctrica Nacional", for the funding and support, M. Eng. Rui Pestana from REN for the advice and collaboration, and Doctor Carlos M. Fonseca for the multi-objective genetic algorithm. The first author thanks the European Commission for the funding through grant PERG-GA-2008-239451.

References

1. Amari, S., Murata, N., Müller, K.R., Finke, M., Yang, H.: Asymptotic statistical theory of overtraining and cross-validation. *IEEE Transactions on Neural Networks* 8(5), 985–996 (1997), doi:10.1109/72.623200
2. Bauer, M., Buchtala, O., Horeis, T., Kern, R., Sick, B., Wagner, R.: Technical data mining with evolutionary radial basis function classifiers. *Applied Soft Computing* 9, 765–774 (2009), doi:10.1016/j.asoc.2008.07.007

3. Billings, S.A., Zheng, G.L.: Radial basis function network configuration using genetic algorithms. *Neural Networks* 8(6), 877–890 (1995)
4. Branke, J.: Evolutionary algorithms for neural network design and training. Tech. Rep. 322, University of Karlsruhe, Institute AIFB, Karlsruhe, Germany (1995)
5. Carse, B., Pipe, A.G., Fogarty, T.C., Hill, T.: Evolving radial basis function neural networks using a genetic algorithm. In: *IEEE International Conference on Evolutionary Computation*, vol. 1, pp. 300–305 (1995), doi:10.1109/ICEC.1995.489163
6. Chen, S., Wu, Y., Luk, B.: Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks. *IEEE Transactions on Neural Networks* 10(5), 1239–1243 (1999)
7. Coello Coello, C.: Recent trends in evolutionary multiobjective optimization. In: Jain, L., Wu, X., Abraham, A., Jain, L., Goldberg, R. (eds.) *Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing*, pp. 7–32. Springer, Heidelberg (2005), http://dx.doi.org/10.1007/1-84628-137-7_2, doi:10.1007/1-84628-137-7-2
8. Coello Coello, C.: Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine* 1(1), 28–36 (2006), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1597059&tag=1, doi:10.1109/MCI.2006.1597059
9. Crispim, E.M., Ferreira, P.M., Ruano, A.E.: Prediction of the solar radiation evolution using computational intelligence techniques and cloudiness indices. *International Journal of Innovative Computing, Information and Control* 4(5), 1121–1133 (2008)
10. Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons., Ltd, Chichester (2001)
11. Ferreira, P.M., Faria, E., Ruano, A.E.: Neural network models in greenhouse air temperature prediction. *Neurocomputing* 43(1-4), 51–75 (2002)
12. Ferreira, P.M., Martins, I.A., Ruano, A.E.: Cloud and clear sky pixel classification in ground-based all-sky hemispherical digital images. In: Ferreira, P.M. (ed.) *Proceedings of CMTEE 2010, the IFAC Conference on Control Methodologies and Technology for Energy Efficiency*. International Federation of Automatic Control, Vilamoura, Portugal (2010)
13. Ferreira, P.M., Ruano, A.E.: Exploiting the separability of linear and non-linear parameters in radial basis function neural networks. In: *IEEE Symposium 2000: Adaptive Systems for Signal Processing, Communications, and Control*, Canada, pp. 321–326 (2000), <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=882493>, doi:10.1109/ASSPCC.2000.882493
14. Ferreira, P.M., Ruano, A.E.: Application of computational intelligence methods to greenhouse environmental modelling. In: (WCCI 2008) *IJCNN 2008 (IEEE World Congress on Computational Intelligence)*. 2008 IEEE International Joint Conference on Neural Networks, pp. 3582–3589 (2008), doi:10.1109/IJCNN.2008.4634310
15. Ferreira, P.M., Ruano, A.E.: On-line sliding-window methods for process model adaptation. *IEEE Transactions on Instrumentation and Measurement* 58(9), 3012–3020 (2009), doi:10.1109/tim.2009.2016818
16. Ferreira, P.M., Ruano, A.E., Fonseca, C.: Genetic assisted selection of rbf model structures for greenhouse inside air temperature prediction. In: *IEEE Conference on Control Applications*, Turkey, pp. 576–581 (2003)
17. Ferreira, P.M., Ruano, A.E., Fonseca, C.: Evolutionary multi-objective design of radial basis function networks for greenhouse environmental control. In: *IFAC World Congress on Automatic Control 16th, Czech Republic* (2005)

18. Ferreira, P.M., Ruano, A.E., Pestana, R.: Improving the identification of rbf predictive models to forecast the portuguese electricity consumption. In: Ferreira, P.M. (ed.) Proceedings of CMTEE 2010, the IFAC Conference on Control Methodologies and Technology for Energy Efficiency. International Federation of Automatic Control, Vilamoura, Portugal (2010)
19. Ferreira, P.M., Ruano, A.E., Pestana, R., Kóczy, L.T.: Evolving rbf predictive models to forecast the portuguese electricity consumption. In: ICONS 2009: The 2nd IFAC Int. Conference on Intelligent Control Systems and Signal Processing, Istanbul, Turkey (2009)
20. Fletcher, R.: Practical Methods of Optimization, 2nd edn. Wiley Interscience, Hoboken (2000)
21. Fonseca, C., Fleming, P.: Non-linear system identification with multiobjective genetic algorithms. In: Proceedings of the 13 IFAC World Congress, vol. C, pp. 187–192 (1996)
22. Fonseca, C., Fleming, P.: Multiobjective optimization and multiple constraint handling with evolutionary algorithms i: A unified formulation. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans* 28(1), 26–37 (1998), doi:10.1109/3468.650319
23. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press, Inc., London (1981)
24. Griffin, I., Fleming, P.: An overview of non-linear identification and control with neural networks. In: Ruano, A.E. (ed.) Intelligent Control Using Soft-Computing Methodologies. Control Series, pp. 89–118. IEE Publishing (2005)
25. Guillén, A., Pomares, H., González, J., Rojas, I., Valenzuela, O., Prieto, B.: Parallel multiobjective memetic rbfns design and feature selection for function approximation problems. *Neurocomputing* 72(16-18), 3541–3555 (2009), doi:10.1016/j.neucom.2008.12.037
26. Haykin, S.: Neural Networks: a Comprehensive Foundation, 2nd edn. Prentice Hall, Inc., Englewood Cliffs (1999)
27. Jung, J., Reggia, J.: Evolutionary design of neural network architectures using a descriptive encoding language. *IEEE Transactions on Evolutionary Computation* 10(6), 676–688 (2006), doi:10.1109/TEVC.2006.872346
28. Kaylani, A., Georgiopoulos, M., Mollaghasemi, M., Anagnostopoulos, G.C., Sentelle, C., Zhong, M.: An adaptive multiobjective approach to evolving art architectures. *IEEE Transactions on Neural Networks* 21(4), 529–550 (2010)
29. Lee, C.W., Shin, Y.C.: Growing radial basis function networks using genetic algorithm and orthogonalization. *International Journal of Innovative Computing, Information and Control* 5(11(A)), 3933–3948 (2009)
30. Leung, C., Lam, F.: Performance analysis for a class of iterative image thresholding algorithms. *Pattern Recognition* 29(9), 1523–1530 (1996)
31. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics* 11(2), 431–441 (1963)
32. McDonnell, J., Waagen, D.: Determining neural network hidden layer size using evolutionary programming. In: Proceedings of the 1993 World Congress on Neural Networks, vol. III, pp. 564–657 (1993)
33. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics* SMC-9(1), 62–66 (1979)
34. Peck, C., Dhawan, A., Meyer, C.: Genetic algorithm based input selection for a neural network function approximator with applications to ssm health monitoring. In: IEEE International Conference on Neural Networks, vol. 2, pp. 1115–1122 (1993), doi:10.1109/ICNN.1993.298714

35. Ridler, T., Calvard, S.: Picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man and Cybernetics SMC-8*(8), 630–632 (1978)
36. Rodríguez-Vázquez, K., Fonseca, C., Fleming, P.: Identifying the structure of nonlinear dynamic systems using multiobjective genetic programming. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans* 34(4), 531–545 (2004), doi:10.1109/TSMCA.2004.826299
37. Ruano, A., Crispim, E., Frazão, P.: Moga design of neural network predictors of inside temperature in public buildings. In: Balas, V., Fodor, J., Várkonyi-Kóczy, A. (eds.) *Soft Computing Based Modeling in Intelligent Systems*. SCI, vol. 196, pp. 35–61. Springer, Heidelberg (2009), doi:10.1007/978-3-642-00448-3-3
38. Ruano, A., Fleming, P., Jones, D.: Connectionist approach to pid autotuning. *IEE Proceedings (part D)*, 139(3), 279–285 (1992), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=141517
39. Ruano, A., Jones, D., Fleming, P.: A new formulation of the learning problem of a neural network controller. In: *Proceedings of the 30th IEEE Conference on Decision and Control*, vol. 1, pp. 865–866 (1991), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=261439, doi:10.1109/CDC.1991.261439
40. Ruano, A.E., Ferreira, P.M., Cabrita, C., Matos, S.: Training neural networks and neuro-fuzzy systems: A unified view. In: *Proceedings of the 15th IFAC World Congress*, vol. 15 (2002)
41. Ruano, A.E., Ferreira, P.M., Fonseca, C.: An overview of non-linear identification and control with neural networks. In: Ruano, A.E. (ed.) *Intelligent Control Using Soft-Computing Methodologies*. Control Series, pp. 37–87. IEE Publishing (2005)
42. Ruano, A.E., Ferreira, P.M., Mendes, H.: Moga design of temperature and relative humidity models for predictive thermal comfort. In: Ferreira, P.M. (ed.) *Proceedings of CMTEE 2010, the IFAC Conference on Control Methodologies and Technology for Energy Efficiency*. International Federation of Automatic Control, Vilamoura, Portugal (2010)
43. Ruano, A.E., Fleming, P., Teixeira, C., Rodríguez-Vázquez, K., Fonseca, C.: Nonlinear identification of aircraft gas-turbine dynamics. *Neurocomputing* 55(3-4), 551–579 (2003), doi:10.1016/S0925-2312(03)00393-X
44. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging* 13(1), 146–165 (2004)
45. Sjöberg, J., Ljung, L.: Overtraining, regularization, and searching for minimum with application to neural networks. In: *Preprint IFAC Symposium on Adaptive Systems in Control and Signal Processing*, pp. 669–674 (1994)
46. Trussel, H.: Comments on picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man and Cybernetics SMC-9*(5), 311 (1979)
47. Yen, G.: Multi-objective evolutionary algorithm for radial basis function neural network design. In: Jin, Y. (ed.) *Multi-objective machine learning*. SCI, vol. 16, pp. 221–239. Springer, Heidelberg (2006), doi:10.1007/3-540-33019-4-10
48. Zitzler, E., Laumanns, M., Bleuler, S.: A tutorial on evolutionary multiobjective optimization. In: Gandibleux, X., et al. (eds.) *Metaheuristics for Multiobjective Optimisation*. Lecture Notes in Economics and Mathematical Systems, vol. 535, pp. 3–37. Springer, Heidelberg (2004)

Structural Learning Model of the Neural Network and Its Application to LEDs Signal Retrofit

Junzo Watada and Shamshul Bahar Yaakob

Abstract. The objective of this research is to realize structural learning within a Boltzmann machine (BM), which enables the effective solution of problems defined in terms of mixed integer quadratic programming. Simulation results show that computation time is up to one fifth faster than conventional BMs. The computational efficiency of the resulting double-layer BM is approximately expressed as the ratio n divided by N , where n denotes the number of selected units (neurons/nodes), and N the total number of units. The double-layer BM is applied to efficiently solve the mean-variance problem using mathematical programming with two objectives: the minimization of risk and the maximization of expected return. Finally, the effectiveness of our method is illustrated by way of a light emitting diodes (LED) signal retrofit example. The double-layer BM enables us to not only obtain a more effective selection of results, but also enhance effective decision making. The results also enable us to reduce the computational overhead, as well as to more easily understand the structure. In other words, decision makers are able to select the best solution given their respective points of view, by means of the alternative solution provided by the proposed method.

Keywords: Structural learning, Hopfield network, Boltzmann machine (BM), double-layer BM, quadratic programming, mean-variance analysis.

Junzo Watada

Graduate School of Information, Production and Systems, 2-7,
Hibikino, Wakamatsu, Fukuoka, 808-0135 Japan

Junzo Watada · Shamshul Bahar Yaakob

School of Electrical Systems Engineering, University Malaysia Perlis,
02600, Perlis, Malaysia

e-mail: junzow@osb.att.ne.jp and shamshul@fuji.waseda.jp

1 Introduction

A neural network consists of a number of mutually connected units (nodes/neurons). The Back-Propagation (BP) neural network has units hierarchically structured, whereas a mutually connected type of neural network has all units connected with each other, and is generally built using large scale data and at the same time requires more computation time and cost spent in execution.

In the case of constructing a hierarchical neural network using BP, the numbers of input and output units are uniquely decided as a function of the numbers of input and output training data, respectively. On the other hand, the numbers of hidden units and hidden layers depend on the learning method as well as on the numbers of input and output units (nodes). There needs to be a minimal number of hidden units (and hidden layers). Generally, such a number is not known a priori. Conventionally, the number of units in the hidden layer is decided by experience. In this case, if we decrease the number of units, computation speed and system cost can both be saved. Also, the BP method depends very much on the initial weight values and it is difficult to forecast an expected value without convergence, even if we select an approximately minimal number of units. In order to overcome such a problem, the network structure is changed recursively and gradually in order to achieve an optimal structure [1, 2]. This process is called structural learning.

In this study, we apply structural learning to a Boltzmann machine. The Hopfield network is an interconnected neural network originally proposed by J.J. Hopfield in 1982 [3]. Now the Hopfield neural network can easily terminate at a local minimum of the describing energy function. The BM [4] is likewise an interconnected neural network, which improves Hopfield network performance by using probabilities to update both the state of a neuron and its energy function, such that the latter rarely falls into a local minimum.

We formulate a two-layered neural network comprising both a Hopfield network and a BM in order to effectively and efficiently select a limited number of units from those available. The Hopfield network is employed in the upper layer to select the limited number of units, and the BM is employed in the lower layer to decide the optimal solution/units from the limited number of units selected by the upper layer. The double-layered BM, whose two layers connect corresponding units in the upper and lower machines, constitutes an effective problem solving method.

Generally speaking, a conventional BM has considerable computational overhead. The reason for this is that the inherent exponential computation time is a function of the number of units. In this study, by building a double-layered BM, both layers are optimally configured by structural learning. The results enable us to reduce the computational time and cost, as well as to more easily understand the internal structure.

In the following sections, the Hopfield network and Boltzmann machine are briefly introduced. Section 3 presents an explanation of the BM approach to mean-variance analysis. Section 4 explains the double-layered BM, followed by simulation results. Section 5 provides overviews of LEDs retrofit. We examine the

effectiveness of the proposed structural learning model for the LEDs retrofit problem in Section 6. In Section 7 we discuss about the simulation and results. Finally, conclusions are drawn in Section 8.

2 Hopfield and Boltzmann Machine

Research into mutually connected network behavior started around 1948. Simply stated, it is difficult to select the required number of units at the same time as minimizing the corresponding energy function. This problem cannot be solved by either Hopfield or Boltzmann neural networks. In the 1970s, some researchers independently came up with the idea of a model of Associative Memory for mutual connected networks [5, 6]. In 1982, Hopfield [3] brought together several earlier ideas concerning recurrent neural networks, accompanied by a complete mathematical analysis. Nowadays, this type of network is generally referred to as a ‘‘Hopfield network’’. Such networks, together with the Back-Propagation algorithm, signaled the re-birth of research into neural networks in the early 1980s, which has continued to the present time. Although the Hopfield network is not a good solution for many applications, it nevertheless warrants revisiting in terms of structure and internal working. This will lead to a modification, by incorporating mutual connections, in order to overcome its drawbacks – in the form of the BM.

The Hopfield network is a fully connected, recurrent neural network, which uses a form of the generalized Hebb rule to store Boolean vectors in its memory. Each unit (neuron)- n has a state value denoted by s_n . In any situation, combining the state of all units leads to a global state for the network. For example, let us consider a network comprising three units s_1 , s_2 and s_3 . The global state at time step t is denoted by a vector s , whose elements are s_1 , s_2 and s_3 . When the user presents the network with an input, the network will retrieve the item in its memory which most closely resembles that particular input.

In general, the Hopfield network operates by taking an input, evaluating the output (in other words the global status s). This global state is the input, providing it works correctly, together with other prototypes, which are stored in the weight matrix by Hebb’s postulate, formulated as

$$w_{ij} = \frac{1}{N} \sum_p X_{i_p} X_{j_p} \quad (1)$$

where, $p = 1 \dots P$, w_{ij} is the weight of the connection from neuron j to neuron i , N is the dimension of the vector, p the number of training patterns, and X_{i_p} the p^{th} input for the neuron i . In other words, using Hebb’s postulate, we create the weight matrix, which stores the entire prototype that we want the network to remember. Because of these features, it is sometimes referred to as an ‘‘Auto-associative Memory’’. However, it is worth noting that the maximum number of prototypes that a Hopfield network can store is only 0.15 times the total number of units in the network [3].

One application of the Hopfield network is to use it as an energy minimizer. This application comes to life because of the ability of Hopfield networks to minimize an energy function during its operation. The simplest form of energy function is given by the following:

$$E = \frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N w_{ji} s_j s_i \quad (2)$$

Here w_{ij} denotes the strength of the influence of neuron j on neuron i . The w_{ij} are created using Hebb's postulate as mentioned above, and they belong to a symmetric matrix with the main diagonal line containing only zeroes (which means there are no self-feedback connections). Because of this useful property, the Hopfield network can also be used to solve combinatorial optimization problems. However, Hopfield networks suffer from a major disadvantage in that they sometimes converge to a local rather than to the global minima, which usually happens when dealing with noisy inputs. In order to overcome this problem, a modification was made to the BM.

The BM is an interconnected neural network, and is a modification of the Hopfield network which helps it to escape from local minima. The main idea is to employ simulated annealing, a technique derived from the metallurgy industry. It works by first relaxing all the particles (in other words, causing them to freely move by applying sufficient "heat"). After that, the temperature is gradually decreased. During this process, the particles will move at lower and lower speed until they are become fixed and form a new structure as the temperature decreases.

Simulated annealing is an optimization technique. In Hopfield nets, local minima are used in a positive way, but in optimization problems, local minima get in the way; one must have a way of escaping from them. When optimizing a very large and complex system (i.e., a system with many degrees-of-freedom), instead of "always" going downhill, we try to go downhill "most of the time". Initially, the probability of not going downhill should be relatively high ("high temperature"), but as time (iterations) go on, this probability should decrease (with the temperature decreasing according to an annealing schedule).

The term "annealing" comes from the technique of hardening a metal (i.e. finding a state of its crystalline lattice that is highly packed) by hammering it while initially very hot, and then again at a succession of decreasing temperatures. It works according to the following algorithm:

1. Pick a unit at random;
2. Compute the probability that the unit should be ON using the formula:

$$p_i(t+1) = S((1/T) \sum_j w_{ij} x_j(t) + b_i)$$
 where $S(x) = 1/(1 + \exp(-x))$ and W is symmetric ($w_{ij} = w_{ji}$);
3. Turn the unit ON with probability $p_i(t+1)$ and OFF with probability $1 - p_i(t+1)$;
4. Decrease the temperature parameter T according to the "annealing schedule".

Now the convergence time of a BM is usually extremely long. According to the "annealing schedule", if T_0 is very large, then a strategy is pursued whereby

neurons are flipping on and off at random, totally ignoring incoming information. If T_0 is close to zero, the network behaves “deterministically”, i.e. like a network of McCulloch-Pitts neurons. Although the way in which a BM works is similar to a Hopfield network, we cannot use Hebb's postulate to create the weight matrix representing the correlations between units. Instead, we have to use a training (learning) algorithm – one based on the Metropolis algorithm.

The BM can be seen as a stochastic, generative counterpart of the Hopfield network. In the BM, probability rules are employed to update the state of neurons and the energy function as follows:

If $V_i(t+1)$ is the output of neuron i , in the subsequent time iteration $t+1$, $V_i(t+1)$ is 1 with probability P , and $V_i(t+1)$ is 0 with probability $1-P$, where

$$P[V_i(t+1)] = f\left(\frac{u_i(t)}{T}\right) \quad (3)$$

Here, $f(\cdot)$ is the sigmoid function, $u_i(t)$ is the total input to neuron i shown in equation (4), and T is the network temperature.

$$u_i(t) = \sum_{j=1}^n w_{ij}V_j(t) + \theta_i \quad (4)$$

where, w_{ij} is the weight between neurons i and j , θ_i is the threshold of neuron i , and V_i is the state of unit i . The energy functions, E , proposed by Hopfield, is written as:

$$E(t) = \frac{1}{2} \sum_{i,j=1}^n w_{ij}V_i(t)V_j(t) - \sum_{i=1}^n \theta_i V_i(t) \quad (5)$$

Hopfield has shown that this energy function simply decreases with learning [3]. There is the possibility that this energy function converges to a local minimum. However, in the case of the BM, the energy function can increase with *minute* probability. Therefore, the energy function will be unlikely to fall into a local minimum. Thus, the combination of Hopfield network and BM offers a solution to overcome the problem of finding the optimal number of units in the neural network. Accordingly, this study proposes a double-layered Boltzmann machine which we discuss in detail in the Section 4.

3 Boltzmann Machine Approach to Mean-Variance Analysis

Mean-variance analysis, originally proposed by H. Markowitz during the early 1950s [7], is a widely used investment theory. It assumes that most decision makers have an aversion to risk even if its obtained return is less. However, it is difficult to identify a utility function because they have different utility structures of their own. Hence, Markowitz formulated mean-variance analysis as the following quadratic programming problem under the restriction that the expected return rate must be more than a certain specified amount.

[Formulation 1]

$$\text{minimize } \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} m_i x_i m_j x_j \quad (6)$$

$$\text{subject to } \sum_{i=1}^n \mu_i m_i x_i \geq R \quad (7)$$

$$\sum_{i=1}^n m_i x_i = 1 \quad (8)$$

$$m_i \in \{0,1\} (i = 1,2,\dots,n) \quad (9)$$

$$x_i \geq 0 (i = 1,2,\dots,n) \quad (10)$$

where R denotes an acceptable least rate of expected return, σ_{ij} a covariance between stock i and stock j , μ_i an expected return rate of stock i , and x_i an investment rate of stock i , respectively.

In Formulation 1, the optimal solution with the least risk is searched under the constraint that the expected return rate should be more than the value a decision-maker arbitrarily gives. The investment rate for each of the stocks is decided for the solution with the least risk to the given expected return rate. Since the risk is estimated under the condition of fixing the rate of the expected return, the decision-maker cannot be fully satisfied with its solution. Therefore, the following Formulation is much more appropriate and reasonable compared with Formulation 1:

[Formulation 2]

$$\text{maximize } \sum_{i=1}^n \mu_i m_i x_i \quad (11)$$

$$\text{minimize } \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} m_i x_i m_j x_j \quad (12)$$

$$\text{subject to } \sum_{i=1}^n m_i x_i = 1 \quad (13)$$

$$\sum_{i=1}^n m_i = S \quad (14)$$

$$m_i \in \{0,1\} (i = 1, 2, \dots, n) \quad (15)$$

$$x_i \geq 0 (i = 1, 2, \dots, n) \quad (16)$$

Formulation 2 is a quadratic programming problem with two objective functions - expected return rate, and degree-of-risk.

Next, we explain how to solve a mean-variance analysis using a BM [8-11]. We transform the mean-variance model described by Formulation 1 or 2 into the BM energy function.

First, we transform the objective function, shown as equation (6), into the following energy function in equation (5) as follows:

$$E = -\frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j \right) \quad (17)$$

In the next step, we show a condition that the total investment rate of all stocks is 1 (note that the investment rate of each stock cannot be less than 0). The condition can be expressed as

$$\left(\sum_{i=1}^n x_i - 1 \right)^2 = 0 \quad (18)$$

Equation (18) can be rewritten as follows

$$\sum_{i=1}^n \sum_{j=1}^n x_i x_j - 2 \sum_{i=1}^n x_i + 1 = 0 \quad (19)$$

Then, as we can transform equation (18) into equation (17), the latter can be rewritten as

$$E = -\frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j + 2 \sum_{i=1}^n \sum_{j=1}^n x_i x_j \right) + 2 \sum_{i=1}^n x_i \quad (20)$$

Finally, we consider the expected return, given by equation (7). Therefore, we can transform equation (7) into equation (21):

$$E = -\frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j + 2 \sum_{i=1}^n \sum_{j=1}^n x_i x_j \right) + 2 \sum_{i=1}^n x_i + K \sum_{i=1}^n \mu_i x_i \quad (21)$$

where K is a real number not less than 0.

If the value K is set to a larger number, the expected return is evaluated much more than the risk. Then, if we determine $K=0$, then the BM converges into a

problem of minimizing its risk. When the energy function of the BM described in this section converges to the global minimum, we can obtain the investment rate of stocks by the output value of each unit. The algorithm of the BM is executed according to the following

[Algorithm 2]

- Step 1. Give the initial value of all units optionally;
- Step 2. Choose a certain unit (i) out of all units at random;
- Step 3. Compute a total of the input $u_i(t)$ into the chosen unit i ($1 \leq i$);
- Step 4. Add a sufficiently small value to the output value $V_i(t+1)$ of unit i , according to the probability P shown in equation (3), and subtract a sufficiently small value from the output value with probability $1-P$. However, the output value is not varied in the case of $u_i(t)=0$;
- Step 5. The output value of units j except i are not varied;
- Step 6. After iterating from Steps 2 to 5, compute the probability of each unit for all units.

4 Double-Layered Boltzmann Machine Example

Conventionally, the number of units is decided on the basis of expert experience. In order to solve this problem, we formulate a double-layered neural network consisting of both Hopfield and Boltzmann neural networks. This double-layered model can be employed to select a limited number of units from those available. The double-layered model has two layers – referred to as the upper and lower layers, respectively. The functions of the layers are as follows:

1. Upper layer (Hopfield neural network) is used to select a limited number of units from the total. This Hopfield layer is called a “supervising layer”.
2. Lower layer (Boltzmann machine) is used to decide the optimal units from the limited number selected in the upper layer. This Boltzmann layer is called an “executing layer”.

This double-layered BM is a new type of neural network model which deletes units (neurons) in the lower layer that are not selected in the upper layer during execution. The lower layer is then restructured using the selected units. Because of this feature, the double-layered BM converges more efficiently than a conventional BM. This is an efficient method for solving a selection problem by transforming its objective function into the energy function, since the Hopfield and Boltzmann networks converge at the minimum point of the energy function.

The double-layered BM just described converts the objective function into energy functions of two components - namely the upper layer (Hopfield network) E_u and the lower layer (Boltzmann machine) E_l , as described below.

Upper layer

$$E_u = -\frac{1}{2} \sum_{i=j}^n \sum_{j=1}^n \sigma_{ij} s_i s_j + K_u \sum_{i=1}^n \mu_i s_i \quad (22)$$

Lower layer

$$E_l = -\frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j + 2 \sum_{i=1}^n \sum_{j=1}^n x_i x_j \right) + 2 \sum_{i=1}^n x_i + K_l \sum_{i=1}^n \mu_i x_i \quad (23)$$

where K_u and K_l are the weights of the expected return rates of the layers, and s_i is the output of the i^{th} unit of the upper-layer.

The double-layered BM is tuned such that the upper layer influences the lower layer with probability 0.9, and the lower layer influences the upper layer with probability 0.1. Thus the double-layered BM is iterated with

$$Y_i = 0.9 y_i + 0.1 x_i$$

for the upper layer, and

$$X_i = x_i (0.9 y_i + 0.1)$$

for the lower layer. Here Y_i in the upper layer is a value transferred to the corresponding nodes in the upper layer, X_i in the lower layer is a value transferred to corresponding nodes in the lower layer, y_i is the value of the present state at node i in the upper layer, and x_i is the value of the present state at node i in the lower layer, respectively.

X_i means that the value is influenced to the tune of 90% from the value of node i in the upper layer. When Y_i is 1, $X_i = x_i$; otherwise, when y_i is 0, 10% of the value of x_i is transferred to the other nodes. On the other hand, Y_i has a 10% influence on the lower layer. Therefore, even if the upper layer converges to a local minimum, the disturbance from the lower layer makes the upper layer escape from this local minimum. When the local minima possess a large barrier, dynamic behavior may be used (by changing 0.9 and 0.1 dynamically) - this phenomenon is similar to simulated annealing.

The algorithm of the double-layered BM is as follows:

[Algorithm 1]

- Step 1. Set each parameter to its initial value.
- Step 2. Input K_u and K_l .
- Step 3. Execute the upper layer.
- Step 4. If the output value of a unit in the upper layer is 1, add some amount of this value to the corresponding unit in the lower layer. Execute the lower layer.

- Step 5. After executing the lower layer at a constant frequency, decrease the temperature.
- Step 6. If the output value is sufficiently large, add a certain amount of the value to the corresponding unit in the upper layer.
- Step 7. Iterate from Step 3 to Step 6 until the temperature reaches the restructuring temperature.
- Step 8. Restructure the lower layer using selected units in the upper layer
- Step 9. Execute the lower layer until reaching the termination condition.

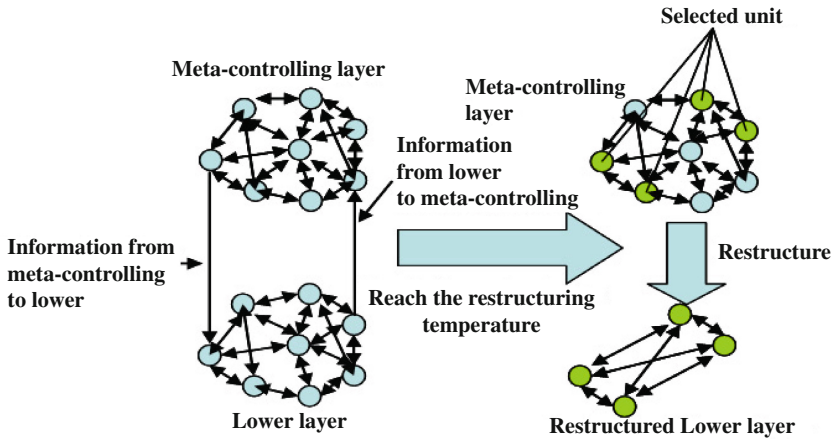


Fig. 1 Meta-controlled BM

5 Overview on LEDs Signal Retrofit

In this study, Traffic signals have been considered a way to improve traffic safety and traffic operations at intersections. Intersections induce more attention for safety analyses than other roadway elements due to the fact many intersections are found to be relatively crash-prone spots from safety point of view. Each year, the number of traffic crashes occurred has been increased. According to the Traffic Safety Facts 2002, in United States, there were 6,316,000 estimated traffic crashes in 2002 [12].

LED lamps have been developed to replace conventional incandescent or fluorescent lamps for increasing reliability and reducing electrical and maintenance costs. The new traffic lights are made out off arrays of LED signals. This is tiny, purely electronic light that is extremely energy efficient and have a very long life. Each LED is about a size of a pencil eraser, so hundreds of them are used together in an array. The LED signals are replacing the conventional incandescent halogen bulbs rated at between 50 and 150 watts. Three main advantages of LED signals:

- a. LED signals are brighter than conventional signals, which enhance intersection safety.
- b. Due to their low wattage, LED signals consume significantly less power, which results in lower energy bills.
- c. LED signals can be expected to run for at least 10 years.

In Japan, there are roughly 2 million traffic signal systems. Japan's government supports 50 percent of changing expenditure for each regional government. Now 10 percent of all systems were already changed to LED type, but the cost of LED type is 1.5 times more expensive than conventional tube type. Japan Economy Newspaper (NIKKEI) on November 21 2007 reported that the traffic accidents were reduced about 30 percent from 2001 to 2005 due to the usage of LED type that has an advantage to eliminate the phantom effect during morning and evening hours. In the same report also indicates that Japan National Police Agency decided to change gradually conventional type traffic signal to LED type for major highway from 2007.

There appears to be growing acceptance of LED signals as viable light sources for traffic signals, and a growing awareness of the potential maintenance and energy savings achievable with LED signals, but many self-governing bodies face significant capital constrains. And huge numbers of intersections could not be replaced simultaneously. These are disincentives within local governments to perform LED retrofits despite their potential life cycle cost benefits. Thus, jurisdictions have to selectively implement LED signals retrofit to enhance traffic security and operate more cost-effectively.

6 LEDs Signal Retrofit and Mean-Variance Problem

Traffic crashes bring out tragic loss of lives, cost many countries tremendous amount of money, and produce substantial congestion to a nation's transportation system. Large percentage of traffic accidents occurs at or near intersections [13]. Pernia indicates that intersections-related crashes make up a very high percentage of the total number of crashes in the roadway system. For example, in the United States, the national statistics show that 22.87 percent of all fatal crashes occurred at intersections or intersection-related locations. Traffic signals have been a way to improve traffic safety and traffic operations at intersections [12].

In order to improve traffic safety and traffic operations at intersections, due to the many advantages of operation and energy consumption, LED traffic light is preferred [14]. LED traffic lights, whose advantages – in comparison to conventional light bulb traffic lights - include the significantly reduction of electric power consumption, the dramatic saving of lower maintenance costs and the improvement of safety due to greater brightness, were so impressive even after a short time that soon further junctions were equipped with LED traffic lights.

Despite their excellent performance, several barriers hinder more rapid retrofit such as high retrofit cost and capital constraints. Thus, one of the most significant challenges in retrofit strategy is to decide which intersection to signalize with LEDs, which to keep on the table. We provide a method for selecting intersections

not through traditional effectiveness measures like cost and performance, but instead through a quantitative analysis of the embedded uncertainty in each potential intersection. Cost and performance in this approach remain central themes in decision making, but uncertainty serves as the focal point to identify potentially powerful combinations of intersections to explore concurrently in decision phases. It presented is a method to identify and quantify uncertainty in intersections, as well as a means to manage it using mean-variance analysis (portfolio theory) and optimization. Perhaps best known to economists and investors, portfolio theory is based on the objective of minimizing risk subject to a decision maker's sufficient return considering his or her risk aversion. This simple concept, as well as the theoretical accuracy that has evolved the theory to practice, is presented as one means of exploring the retrofit strategy of potential intersections around the central theme of uncertainty.

A mean-variance approach is proposed to change the situation of investing a large amount of money on maintenance and repair based by accident rates. We intend to invest using the frequency of accidents under the consideration of past data.

Portfolio theory treats a mathematical allocation problem of a given amount of money among several different available investments, such as stocks, bonds and securities. This is named the portfolio selection problem. Markowitz originally proposed and formulated the mean-variance approach based on the portfolio selection problem [7], [11]. That is, assuming the time series of return rates, the theoretical method enables us to determine the highest investment rate, which minimizes the risk or variance of profit, affirming the highest rate of the expected return which a decision maker expects. This method is formulated as a quadratic programming. In this paper, a portfolio selection problem is formulated as a mathematical programming with two objectives to minimize risk and maximize the expected return, since the efficient frontier should be considered in the discussion of a portfolio selection. Yang et al. 2004 proposed the multi-objective programming model of portfolio and compute the optimal solution with some methods by a neural network [15].

A Hopfield network and a Boltzmann machine are used to find an optimal solution [16], [17]. In this paper, we applied the concepts of a Boltzmann machine to solve the portfolio selection problem efficiently. The Boltzmann machine [4] is an interconnected neural network proposed by G. E. Hinton. The Boltzmann machine is a model that improves a Hopfield network using probability rule to update the state of a neuron and its energy function. Thus, the energy function of the Boltzmann machine hardly falls into a local minimum. For that reason, if we transform the objective function of a portfolio selection problem into an energy function of the Boltzmann machine, it enables us to solve the portfolio selection problem as its highly approximate solution. And then, the output value of each unit represents the investing rate to each stock. In the conventional method to solve portfolio selections, the investing rate to each stock is decided to realize the minimum risk under the constraints that the goal rate of an expected return given by a decision maker should be guaranteed. But in this proposed method, the

objective of solving a portfolio selection problem is not only to minimize its risk but also to maximize the expected return rate. Therefore, the Boltzmann machine can provide the investment rate for each intersection using the output of each unit of the neural network. Retrofit traffic signals investment problems are presented in order to demonstrate the effectiveness of our proposal.

7 Numerical Example of LEDs Signal Retrofit

We take 10 intersections in Hiroshima with their 8-years accident numbers into account, in which portfolios of retrofit can be analyzed and optimized. It proposed is an effective retrofit strategy where risk, measured by the variance in accident numbers, is considered together with accident mean. Our analysis of trade-off between accident numbers means and variance employs mean-variance analysis and meta-controlled Boltzmann machine, which are considerably efficient as the number of intersections dramatically increases. It seemed obviously that investors are concerned with risk and return, and that these should be measured for the portfolio as a whole. Variance (or, equivalently, standard deviation), came to mind as a measure of risk of the portfolio. The fact that the variance of the portfolio, that is the variance of a weighted sum, involved all covariance terms added to the plausibility of the approach. Since there were two criteria - expected return and risk - the natural approach for an economics program was to imagine the investor selecting a point from the set of optimal expected return, variance of return combinations, now known as the efficient frontier. In this section, we employ meta-controlled Boltzmann machine as an efficient model to solve this trade off.

The simulation parameters employed are in the following step:

Upper layer -

1. The change is done with 0.001 interarrival temperatures.
2. Each unit is set to an initial value of 0.1.
3. The constant K is simulated for 0.0, 0.3, 0.5, 1.0 and 2.0.

Lower layer -

1. The temperature T of the BM is changed from 100 to 0.0001.
2. The change is done with 0.001 interarrival temperatures.
3. The initial setting for each unit is 0.1.
4. The constant K is simulated for 0.0, 0.3, 0.5, 1.0 and 2.0.
5. As the BM behaves probabilistically, the result is taken to be the average of the last 10, 000 times.

The implementation procedure of the proposed method is described as in the following five steps:

Step	Task
Step 1	Identifying the right uncertainty - Security
Step 2	Quantifying individual uncertainties - Accident Numbers
Step 3	Postprocessing the uncertainty - Covariance Matrix
Step 4	Implementing Portfolio Theory - Mean-variance Analysis
Step 5	Determining the optimal maintenance strategy - Selection of K

Step 1. Identifying the Right Uncertainty

Identifying the right uncertainties is the first step in mean-variance analysis. The right uncertainty will have the following characteristics. The characteristic of an uncertainty that should be included in the analysis is one that differentiates one asset from another. An example of this characteristic can be found in a set of intersections that they don't rely on the same security. For example, accidents occurred in Intersection1 by a mean number of 21, but Intersection 9 by a number of 12. Security is just one source of differentiating uncertainty, policy, market conditions or manufacturing capability are others. In our case, we select 10 intersections around Fukuyama station as shown in Fig. 2.

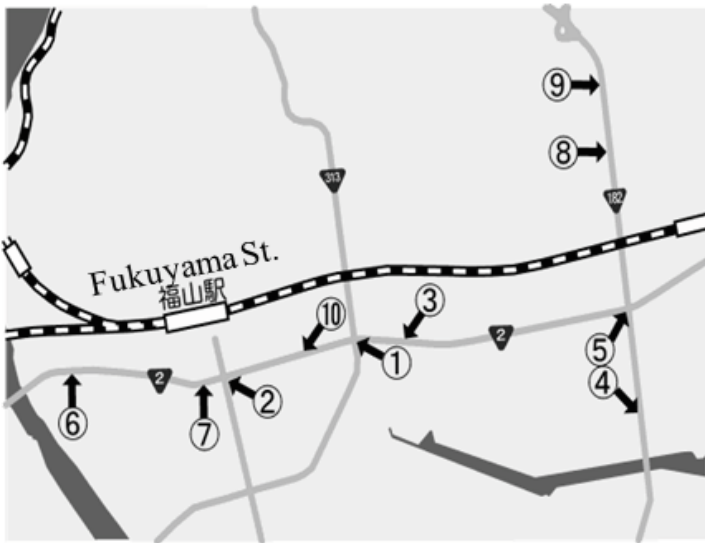


Fig. 2 Location of intersections

Step 2. Quantifying Individual Uncertainties

Once the relevant sources of uncertainty have been identified, the next step is to apply some level of probability and impact to them. Some individual uncertainties can be very straightforward to quantify. For example, if the security model being used is based on the historical data, this model typically has standard deviations that can be included as security modeling uncertainty. Uncertainty identified by security is quantified based on 8-years accident numbers. Table 1 shows historical accident numbers of 10 intersections as shown in Fig. 2. Other uncertainties might not be so straightforward to quantify. These could arise from market conditions, policy uncertainty, new technology or novel architectural concepts.

Table 1 Accident number in intersections

	1998	1999	2000	2001	2002	2003	2004	2005	Mean
Int.1	18	28	21	17	18	22	20	24	21.0000
Int.2	20	14	17	20	15	24	18	22	18.7500
Int.3	13	19	20	13	20	14	17	14	16.2500
Int.4	19	20	17	12	16	18	16	13	16.3750
Int.5	19	12	11	19	17	13	10	12	14.1250
Int.6	11	19	10	9	14	11	16	10	12.5000
Int.7	10	14	8	12	10	4	7	9	9.2500
Int.8	15	17	16	7	11	17	15	13	13.8750
Int.9	13	15	7	9	15	14	11	12	12.0000
Int.10	10	6	13	11	9	10	15	12	10.7500

Step 3. Post-processing the Uncertainties

Once the uncertainties have been quantified for each alternative, it is necessary to post-process and feed the data to the next step in the approach, mean-variance analysis. At this point, statistics of each distribution should be calculated. This includes standard measures of expected value and standard deviation or variance.

Once individual distributions have been investigated, the set of distributions also needs to be post processed to develop the covariance matrices for use in implementing the portfolio optimization. The covariance matrix represents the relative independence of the assets, as well as the uncertainty of the assets. The matrix is created, as shown in Fig. 3, by placing the variance of assets on the diagonal and using pair-wise covariance, as calculated in (24), on the off-diagonals.

$$\sigma_{x_1, x_2} = \rho_{x_1, x_2} \sigma_{x_1} \sigma_{x_2} \quad (24)$$

<i>Assets</i>	X_1	X_2	X_3	• •	X_n
X_1	σ_1^2	$\rho_{12}\sigma_1\sigma_2$	$\rho_{13}\sigma_1\sigma_3$	• • •	
X_2	$\rho_{12}\sigma_1\sigma_2$	σ_2^2	$\rho_{23}\sigma_2\sigma_3$	• • •	
X_3	$\rho_{13}\sigma_1\sigma_3$	$\rho_{23}\sigma_2\sigma_3$	σ_3^2	• • •	
•	•	•	•	• • •	
•	•	•	•	• • •	
X_n	•	•	•	• • •	

Fig. 3 Covariance matrix

Table 2 shows the covariance matrix of 10 intersections that used in this case study.

Table 2 Covariance matrix of 10 intersections

	Int.1	Int.2	Int.3	Int.4	Int.5	Int.6	Int.7	Int.8	Int.9	Int.10
Int.1	11.750	-1.875	2.625	3.500	-7.625	5.875	1.750	6.750	3.000	-3.500
Int.2	-1.875	10.188	-7.688	-2.656	0.906	-7.125	-5.688	-0.281	-1.000	2.688
Int.3	2.625	-7.688	8.438	2.531	-4.281	5.250	1.313	2.531	0.250	-0.938
Int.4	3.500	-2.656	2.531	6.734	-1.547	4.688	-0.094	6.797	3.250	-3.156
Int.5	-7.625	0.906	-4.281	-1.547	11.609	-4.188	3.594	-6.984	1.250	-2.969
Int.6	5.875	-7.125	5.250	4.688	-4.188	10.750	3.125	4.313	4.875	-3.250
Int.7	1.750	-5.688	1.313	-0.094	3.594	3.125	8.188	-3.344	0.875	-4.063
Int.8	6.750	-0.281	2.531	6.797	-6.984	4.313	-3.344	10.359	2.125	-0.781
Int.9	3.000	-1.000	0.250	3.250	1.250	4.875	0.875	2.125	7.250	-4.750
Int.10	-3.500	2.688	-0.938	-3.156	-2.969	-3.250	-4.063	-0.781	-4.750	6.438

Step 4. Applied Portfolio Theory

In order to enable the decision maker implement LED signals retrofit at an optimal set of assets to pursue that maximize return while at the same time consider his aversion to risk. The specific class of optimization is a quadratic optimization based on an appropriate balance of risk and returns. These risks and returns are typically derived from historical accident numbers in intersections historically. The quadratic programming problem can be solved by a method named mean-variance analysis employed meta-controlled Boltzmann machine efficiently.

Table 3 Result of simulation in investment rate for each intersection

	K=0.3	K=0.5	K=0.7	K=1.0
Int.1	0.281	0.287	0.289	0.289
Int.2	0.224	0.244	0.245	0.251
Int.3	0.237	0.239	0.242	0.245
Int.4	-	-	0.017	0.045
Int.5	0.255	0.197	0.155	0.041
Int.6	-	-	-	0.027
Int.7	-	-	-	-
Int.8	-	-	-	-
Int.9	-	-	-	-
Int.10	-	0.031	0.051	0.101

As shown in Table 3, in case of $K = 0.3$, from the total budget allocated, Intersection1 should be invested in by 28.1 percent, Intersection2 25.8 percent, Intersection3 23.7 percent and Intersection5 25.5 percent. Other intersections which are not included in the list of units after restructuring will not get any investment. So in the case of $K = 0.3$, we can select Intersection1, Intersection2, Intersection3, and Intersection5 in order to invest maintenance cost in by 28.1 percent, percent, 22.4 percent, 23.7 percent and 25.5 percent. In case of $K = 0.5$, five intersections were selected in the list of units after restructuring. There were Intersection1, Intersection2, Intersection3, Intersection5 and Intersection10 with investment maintenance cost percentage 28.7 percent, 24.4 percent, 23.9 percent, 19.7 percent and 3.1 percent. In case of $K = 0.7$, six intersections were selected and in case of $K = 1.0$, seven intersections were selected in the list of units after restructuring. From that, we can conclude that the number of selected wards in the restructured list is directly proportional to K .

Step 5. Determining the Optimal Maintenance Strategy

In Step 4, the portfolio theory algorithm is developed and a mean-variance analysis employed neural network is designed that shows the set of solutions on the efficient frontier from which an optimal solution should be chosen. In order to determine where a decision maker’s optimal strategy lies, their level of aversion to uncertainty must be quantified. The most straightforward method of calculating a decision maker’s aversion is to find an indifferent curve between the value and uncertainty that accurately reflects his/her interests.

Table 4 Expected investment rate and risk

K	Return Rate	Risk
0.3	16.163	0.036
0.5	16.261	0.046
0.7	16.311	0.051
1.0	16.353	0.059

The expected investment rate and risk are calculated, as shown in Table 4 and also indicates four different levels of risk aversion, value of K , reflect the decision maker's different preference. When K is set at a larger value, the solution is obtained with high investment rates and high risk.

Table 5 Comparison of conventional BM and meta-controlled BM

No. of Intersection	Computational Times(sec)	
	Conventional Boltzmann Machine	Meta-controlled Boltzmann Machine
10	7.21	6.42
40	12.11	8.52
160	43.41	12.61
640	219.12	31.90

Table 5 compares meta-controlled Boltzmann machine and conventional Boltzmann machine, employing various sizes from 10 intersections to 640 intersections. The computing time of the meta-controlled Boltzmann machine is drastically shorter than a conventional Boltzmann machine. The reason for this is because the meta-controlled Boltzmann machine deletes useless units during the restructuring step. By contrast, a conventional Boltzmann machine computes all units until the termination condition reached.

8 Conclusions

This paper demonstrates that structural learning with a double-layer Boltzmann machine has various advantages. Structural learning is employed to decide the

optimal number of hidden units of the neural network, and its appropriateness has been verified. As a result, it was shown that structural learning as proposed in this paper can successfully determine the optimal substation solution, as illustrated in the numerical example. The simulation also showed that computational times are significantly decreased compared with a conventional BM.

This paper demonstrates that proposed method has various advantages. Mean-variance analysis and structural learning are employed to solve the problem on how to choose intersections to implement LED signal retrofit, and its appropriateness has been verified. As a result, it was shown that the proposed method in this paper can successfully determine the optimal intersection solution, as illustrated in the numerical example. The simulation also showed that computational times are significantly decreased compared with a conventional Boltzmann machine.

Mean-variance analysis, which employs the portfolio method using a meta-controlled Boltzmann machine, can deal much more effectively with these types of problems. The results obtained show that the selection, investment expense rate of intersections, and reduced computation time can be prolonged to increase cost savings. The results also demonstrate that our proposal for incorporating structural learning into the Boltzmann machine is effective and can enhance the decision making process.

References

1. Asahi T., Murakami K., Sagamihara T.: BP algorithm for optimality and highly speeding of neural network structural learning, Report of Institute of Electronics, Information NC90-64 (1991)
2. Matsunaga, Y., Nakade, Y., Nakagawa, O., Kawanaka, M.: Back propagation algorithm to auto eliminate redundant hidden units from competition. *Trans. Institute of Electronics, Information J79-D-II(3)*, 403–412 (1996)
3. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proceedings of National Academy Science* 79, 2554–2558 (1982)
4. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for Boltzmann Machines. *Cognitive Science* 9, 147–169 (1985)
5. Kohonen, T., Oja, E.: Fast adaptive formation of orthogonalizing filters and associative memory in recurrent networks of neuron-like elements. *Biological Cybernetics* 21(2), 85–95 (1976)
6. Nakano, K.: A model of associative memory. *IEEE Trans. Systems, Man and Cybernetics* 2(72), 380–388 (1972)
7. Markowitz, H.: *Mean-Variance analysis in portfolio choice and capital Markets*. Blackwell, Malden (1987)
8. Watada, J.: Fuzzy portfolio selection and its applications to decision making. *Tatra Moutains Math. Publ.* 13, 219–248 (1997)
9. Watada, J., Oda, K.: Formulation of a two-layered Boltzmann Machine for portfolio selection. *Intl. J. Fuzzy Systems* 2(1), 39–44 (2000)

10. Watanabe, T., Watada, J., Oda, K.: Hierarchical Decision Making in Strategic Investment by a Boltzmann Machine. *Intl. J. Uncertainty, Fuzziness and Knowledge-Based Systems* 7(4), 429–437 (1999)
11. Yaakob, S.B., Takahashi, T., Okamoto, T., Tanaka, T., Minh, T.D., Watada, J., Xiaojun, Z.: Structural learning approach to replacing unreliable units in a power system. In: *Proc. of 2008 Intl. Conf. on Condition Monitoring and Diagnosis, CMD 2008*, art. No. 4580352, pp. 570–575 (2006)
12. Pernia, J., Lu, J.J., Zhuo, Y., Snyder, D.: Effects of traffic signal installation on intersection crashes. *Advances in Transportation Studies an International Journal Section B 2*, 83–95 (2004)
13. Yong-Kul, K., Dong-Young, L.: A traffic accident recording and reporting model at intersections. *IEEE Transactions on Intelligent Transportation Systems* 8(2), 188–194 (2007)
14. Traffic Eng. Div., Dept. of Public Works, City of Little Rock, Arkansas, US: Conventional vs LED traffic signals; operational characteristics and economic feasibility, Final Report Project Sponsored by Arkansas Dept. of Economic Development (2003)
15. Yang, Y., Cao, J., Zhu, D.: A study of portfolio investment decision method based on neural network. In: Yin, F.-L., Wang, J., Guo, C. (eds.) *ISNN 2004. LNCS*, vol. 3174, pp. 976–981. Springer, Heidelberg (2004)
16. Jurado, F.J.S., Penas, M.S.: Hopfield Neural Network and Boltzmann Machine Applied to Hardware Resource Distribution on Chips. In: Mira, J., Álvarez, J.R. (eds.) *IWINAC 2007. LNCS*, vol. 4527, pp. 387–396. Springer, Heidelberg (2007)
17. Yaakob, S.B., Watada, J.: Maintenance of a traffic lights based on portfolio model. *IJSSST* 9(2), 11–22 (2008)

Shamshul Bahar YAAKOB



He received a degree in Computer Eng. in 1991 from Shizuoka University, Japan and an MEng. Degree in Electrical and Electronic System in 1996 from Nagaoka University of Technology (NUT), Japan. He is currently a lecturer in School of Electrical System Engineering, Universiti Malaysia Perlis, Perlis, Malaysia. He is pursuing a Ph.D degree at Waseda University. His research interests include soft computing, reliability optimization and multi criterion optimization.

Junzo WATADA



He received his B.S. and M.S. degrees in electrical engineering from Osaka City University, Japan, and his Ph.D degree from Osaka Prefecture University, Japan. Currently, he is a professor of Management Engineering, Knowledge Engineering and Soft Computing at Graduate School of Information, Production & Systems, Waseda University, Japan. He was a recipient of Henri Coanda Gold Medal Award from Inventico in Romania in 2002 and a fellow of both SOFT and BMFSA. He is a contributing principal editor, a co-editor and an associate editor for various international journals, including JSCE of IMECH E and IJBSCHS. His professional interests include soft computing, tracking system, knowledge engineering and management engineering.

Robustness of DNA-Based Clustering

Rohani Abu Bakar, Chu Yu-Yi, and Junzo Watada

Abstract. The primary objective of clustering is to discover a structure in the data by forming some number of clusters or groups. In order to achieve optimal clustering results in current soft computing approaches, two fundamental questions should be considered; (i) how many clusters should be actually presented in the given data, and (ii) how real or good the clustering itself is. Based on these two fundamental questions, almost clustering method needs to determine the number of clusters. Yet, it is difficult to determine an optimal number of a cluster group should be obtained for each data set. Hence, DNA-based clustering algorithms were proposed to solve clustering problem without considering any preliminary parameters such as a number of clusters, iteration and, etc..

Because of the nature of processes between DNA-based solutions with a silicon-based solution, the evaluation of obtained results from DNA-based clustering is critical to be conducted. It is to ensure that the obtained results from this proposal can be accepted as well as other soft computing techniques. Thus, this study proposes two different techniques to evaluate the DNA-based clustering algorithms either it can be accepted as other soft computing techniques or the results that obtained from DNA-based clustering are not reliable for employed.

1 DNA Computing Methods for Solving Clustering Problems

DNA computing began in 1994 when Leonard Adleman has first shown that computing can be done using DNA to solve one of NP-Complete problem Hamiltonian

Rohani Abu Bakar
02600, Perlis, Malaysia, School of Electrical Systems Engineering,
University Malaysia Perlis
e-mail: rohani.abu.bakar@gmail.com

Chu Yu-Yi . Junzo Watada
2-7, Hibikino, Wakamatsu, Fukuoka, 808-0135 Japan, Graduate School of Information,
Production and Systems, Waseda university
e-mail: bolero168@hotmail.com, wagtada@waseda.jp

path problem (HPP) and obtained solutions using DNA experiments. Biomedical moleculars have been applied to solve kind of computing problems [1].

Clustering is regarded as a consortium concept combines algorithms that aims to reveal a structure in high dimensional data and obtain the collections of meaningful relationships in data and information granules. DNA computing is used to supports as the development of clustering technique. This approach is particular when dealing with huge data sets with the unknown number of clusters and encountering a heterogeneous character of available data.

The primary objective of clustering is to discover the structure in a data by forming some number of clusters or groups. We expect that similar objects or patterns will be placed in the same cluster while different objects are assigned to different clusters. Clustering is widely used in various areas such as machine learning, image analysis, data mining, bioinformatics and etc. particularly in dealing with a very large database.

Currently a great deal of application in proximity algorithms to cluster different sources of information, and it involves specific field knowledge pertinent to proceeding problems. For instance, Oehler and Gray have developed a clustering technique for solving compression performance problems in signal processing and vector quantization [4]. Shopbell *et al.* has proposed a clustering technique to cluster around objects in sky for astronomy [5]. Jiang and Tuzhillin focused on clustering customer's interests studying in relation with a certain marketing problem [6]. Jimmy *et al.* addressed several issues in clustering medical data [7].

The ultimate challenge of clustering associates with a combinatorial explosion of the search space. Another challenge comes with the fact that most clustering techniques require the number of clusters is provide in advance. Meanwhile, a number of enhancement of the generic clustering techniques; such as tabu search and simulated annealing; have been developed. Those refinement are typically restricted to small data sets only [8].

In this study, the researchers face substantial combinatorial variant that they have to deal with a significant number of clusters in order to optimize clustering processes. Though it is not impossible, handling a combinatorial issue as mentioned above using conventional computing technique is extremenely difficult. A new approach thatcan help to reduce the processing time as well as the memory requiments without computing arises as one of the viable and computationally alternative.

2 Background Study of Clustering Problems

Clustering is concerned with grouping a collection of patterns (observations, data items, or feature vectors) into groups (clusters). Clustering has been addressed by researchers in various disciplines; this reflects its broad appeal and usefulness as one of the fundamental mechanisms of exploratory data analysis. The advantages of clustering algorithms are enormous.

Each of the clustering algorithms comes with some underlying rationales and offers certain insight into the data. Additionally, each clustering algorithm comes with its own underlying optimization scheme, validation tools and computational

enhancements. It depends on the specific category of problems, different terminology, notation, assumptions and resulting outcomes have been encountered. One may refer to several clustering techniques including fuzzy clustering [9], the use of stochastic complexity functions [10], statistical clustering [11] and clustering is done based on the principle of curve technique [12] and etc.

As pointed out by Jain *et al.* [13], a majority of these approaches and algorithms proposed in the literature are unable to handle large data set. While the world witnesses, a growing diversity and a surprising sophistication of clustering methods; such as those based on genetic algorithms, tabu search and simulated annealing, all these methods are typically restricted to relatively small data sets. In order to support the claim, Table 1 (refer to [8]) presents time and space complexity of several well-known clustering algorithms. Referring to the table, n represents the number of patterns, k represents the number of clusters and l denotes the number of iterations used by the algorithm.

Most of these clustering algorithms exhibit polynomial or exponential complexity. The problem becomes even more challenging if the number of clusters is unknown [8] and has to be identified. These situations apparently show that the DNA computing can emerge as an interesting and viable alternative in computing field.

Table 1 Complexity of selected clustering algorithms cited from [12]

Clustering Algorithm	Time Complexity
k -means	$O(n^{dk+1} \log n)$
K-median	$O(ndk + 2^{(k/\epsilon)O(1)} d^2 n^\sigma)$
Single-line	$O(n^2 \log n)$
Complete-line	$O(n^2 \log n)$
Hierarchical agglomerative algorithms	$O(n^2 \log n)$
Minimal spanning tree (MST)	$O(n^2 \log n)$

With regards to DNA, clustering method is widely employed in the genome database. A lot of techniques have been proposed to cluster around genome sequences and DNA micro arrays; such as gene cluster based on the most similarity tree (CMST) as has been proposed by Lu *et al.*[11]. The basic idea of this method is to express elements in set G , which referred as set of genes in this study. The DNA sequences in a cluster are included as a subset of elements in the set G itself. The main advantage of this algorithm is that the number of cluster is customized at the end of processing, and it is not fixed as a k -means and self-organization map (SOM).

Volfovsky *et al.* [14] has developed clustering methods based on repeated analysis of DNA sequences. This method has been proposed to distribute DNA sequences into different clusters according to similar condition. The transcription start site (TSS) method has been proposed by FitzGerald *et al.*[15] to cluster around DNA sequences in the human promoter database. However, this method is unable to identify a single DNA sequence that is relatively clustered to the TSS for the majority of promoters. Sang *et al.*[16] have developed software named

CLAGen for clustering and annotating gene sequences. In their tool, they implemented a suffix tree algorithm. Their study found that, CLAGen has been successfully evaluated with 42 gene sequences in a TCA cycle of bacteria to 11 clusters. The study has also proven that the method can find the longest subsequence of each cluster.

Joseph *et al.* [17] has proposed to improve the hierarchical algorithm in order to cluster around a gene. The method was named as optimal linear leaf ordering of trees. Joseph *et al.* [17] tested the effectiveness of their algorithm on several types of data sets, which obtained from randomly generated data set, generated data set and an artificial data set. Based on the experiment, they found that the proposed clustering algorithm exhibited $O(n^4)$ time complexity of the original hierarchical algorithm in the study was $O(n^3)$, which is better algorithm, but, they believed that the latest proposed algorithm is most practical to apply because the time consumption of the algorithm is very reasonable as compared to the one of an algorithm constructed from the tree approach.

Other than that, Kim *et al.* [18] suggested to consider the use of the Fuzzy C-Means (FMC) in overcoming possible limitations of binary {0-1} clustering. In the suggestion, once the smallest value is determined, its value and transformed location are used for normalizing micro array data set as well as generalizing data set for simulation purposes. The experimental results have shown that scale of lower normalization is more robust when clustering gene from general micro array data than the two commonly used scale and location adjustment methods in particular when dealing with samples that exhibit changing expression patterns or when some noise is inserted.

This chapter also exposes two different methods based on DNA computing that are proposed for solving clustering problem. The first algorithm is based on mutual order distance meanwhile the second algorithm is based on proximity approach. At the end of this chapter, the details of each proposed algorithm will be discussed.

3 Robustness of DNA-Based Clustering Algorithms

The primary objective of clustering is to discover a structure in data by forming some number of clusters or groups. Any clustering techniques evolve a $K \times n$ partition matrix $U(X)$ of a data set X , assuming that $X = \{x_1, x_2, \dots, x_n\}$, where each x_i is an element in R^m , is partitioned into the number, say K , of clusters (C_1, C_2, \dots, C_K). The partition matrix $U(X)$ is represented as $U = [u_{kj}]$, $k=1, \dots, K$, and $j=1, \dots, n$, where u_{kj} is the membership of pattern x_j to cluster around C_k . In crisp partitioning, the following condition holds: $u_{kj} = 1$ if $x_j \in C_k$; otherwise, $u_{kj} = 0$. In general, clustering can be categorized broadly into two classes; partitioning and hierarchical clustering.

Maulik and Bandyopadhyay address two fundamental questions that should be considered in any typical clustering solution;

How many clusters should be actually presented in the given data?

How real or good the clustering itself is?

Based on these two questions, clustering method needs to determine the number of clusters and also the goodness or validity of the obtained clusters.

Additionally, robustness in executing clustering algorithm is another critical issue that has been discussed in these years among clustering researchers. Robustness or stability of clustering algorithm is essentially measured by two aspects;

- i. Evaluation of changes in input data due to errors; and
- ii. The measurements of the differences in the resulting classification.

Robustness can be defined as the ability of the systems or algorithms to handle stresses, pressures or changes in procedure or circumstance. In other words, robustness deals with the extent of a system or component to perform appropriately in the presence of invalid inputs or stressful environmental conditions.

In soft computing, an algorithm can be considered as robust if it continues to operate despite abnormalities in input, calculations. Formal technique, such as fuzz testing, is essential in providing robustness, as this type of testing involves invalid or unexpected inputs into the algorithms. Fuzz testing is a software testing technique that applies invalid, unexpected, or random data to the inputs of program. Using this testing, if the program stops working (by crashing or failing built-in code assertions) the defect can be noted. In clustering research area, issue of robustness in executing clustering algorithm is very critical issue that has been discussed over years.

To overcome this problem, Yu has suggested that three distortion types are; (i) small-sample effects; (ii) the presence of runs in the sample; (iii) the presence of Markov type dependence of class indexes. There are several methods for measuring the robustness or stability of proposed clustering algorithms. For instance, Rand has proposed a method to measure the similarity between two different clusters or partitions of the same set of an object.

In Rand's proposal, the measurement essentially considers how each pair of objects is assigned to cluster up in two different classifications. If any pair of the objects is placed together in some cluster in each of the classifications, or if the objects in the pairs are assigned to different clusters in both classifications, then this pair of objects is said to be similarly placed. In contrast, an object pair is defined to be differently placed if the pair is in the same cluster in one classification and in different clusters for the other.

Therefore, the objective of this chapter was to evaluate the robustness performance of two different DNA-based clustering algorithms that have been previously proposed. In evaluating the robustness of these proposed algorithms, three different artificial data sets were employed. Similarly, a distance measurement was employed to observe the different between two sets of a result (i.e., before and after some noise is inserted). Furthermore, three different conventional clustering, namely as k-means, Fuzzy C-Means and Gustoffan Kessel algorithms and two validity indices were considered in this paper as a comparison with both DNA-based clustering algorithms.

4 Proximity Distance Approach

Assuming that a certain sample or pattern is denoted by x_i , where $i = 1, \dots, n$. On the other hand, when dealing with vectors, it is typically denoted using boldface, say x, y or l, \dots, n characterized by m features $x_{ij}, j = 1, \dots, m$. The distance between two patterns x_α and $x_\beta, D(x_\alpha, x_\beta)$, comes in the form of :

$$D(x_\alpha, x_\beta) = \sqrt{\sum_{j=1}^m (x_{\alpha j} + x_{\beta j})^2} \quad (1)$$

The mutual distances between patterns are organized in the matrix form Δ :

$$\Delta = \begin{bmatrix} 0 & D(x_1, x_2) & D(x_1, x_3) & \cdots & D(x_1, x_n) \\ D(x_2, x_1) & 0 & D(x_2, x_3) & \cdots & D(x_2, x_n) \\ D(x_3, x_1) & \vdots & 0 & \cdots & D(x_3, x_n) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ D(x_n, x_1) & D(x_n, x_2) & D(x_n, x_3) & \cdots & 0 \end{bmatrix} \quad (2)$$

Given the above matrix, the distances between successive pairs of patterns are afterwards transformed into the corresponding order values denoted here by $O(x_\alpha, x_\beta)$. In essence, the distances are ordered and assigned to the corresponding integer values. These values range from 1 to largest value as $\frac{n(n-1)}{2}$.

The matrix arrangement is shown as follows:

$$\Delta = \begin{bmatrix} 0 & O(x_1, x_2) & O(x_1, x_3) & \cdots & O(x_1, x_n) \\ O(x_2, x_1) & 0 & O(x_2, x_3) & \cdots & O(x_2, x_n) \\ O(x_3, x_1) & \vdots & 0 & \cdots & O(x_3, x_n) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ O(x_n, x_1) & O(x_n, x_2) & O(x_n, x_3) & \cdots & 0 \end{bmatrix} \quad (3)$$

C_β Code	x_α Code	Ordered distance $O(x_\alpha, C_\beta)$	C_β Code
----------------	-----------------	---	----------------

Fig. 1 DNA encoding design for candidates of centre for each cluster group

Some patterns were selected to serve as centres of the clusters, therefore, K clusters would be formed and they were considered clusters C_1, \dots, C_K , respectively. At the early stage of the process, all patterns could serve as possible

candidates for centre of the corresponding groups (clusters). Based on the assumption, the corresponding DNA coding scheme appears as illustrated in Figure 1.

Code C_β represents a sequence for potential centre pattern β . Meanwhile x_α represents a pattern that may become a member of cluster β , Order $O(x_\alpha, C_\beta)$ is a sequence that represents an ordered distance between pattern x_β and pattern x_α and the sequence is ended by a sequence that represents C_β .

In order to differentiate the candidate of the centre of each cluster and the member of the cluster group, a specific nucleotide was embedded when coding the candidate centre. This specific nucleotide is named as *mark*. The function of this *mark* is similar to its function in mutual distance approach. At the same time, all patterns can serve as a member of a supposed cluster and all patterns need to be represented by a DNA sequence as shown in Figure 2.

C_β Code	SpecialMarking Code	x_α Code	OrderedDistance $O(x_\alpha, C_\beta)$	C_β Code
----------------	---------------------	-----------------	--	----------------

Fig. 2 DNA encoding

The coding scheme realized for members of cluster is started with candidate of centre (C_β code) sequence with certain length, followed by (x_α code) code for pattern itself, followed by ordered distance between pattern of candidate of centre ($O(x_\alpha, x_\beta)$) and itself. Finally ends up with the sequence for candidate of centre (C_β code) once more.

For both design schemes, 8-mer of nucleotides is employed to represent the pattern and special *marking* code. However, the specific DNA sequence with different repeat numbers is employed to represent the ordered distance between pattern of centre and pattern itself. Consequently, the total length of DNA sequence for each pattern is different regarding to the different length of their distances. If two patterns are separated with large distance, they may contain large difference number of their DNA sequence. In case that the distance between 2 patterns is short, the length of DNA code is shorter since fewer times of its DNA sequence repeat of order distance part.

According to the definition introduced above, the following procedures are considered to identify k clusters with their n patterns; All designed DNA sequences are placed into a test tube, which is marked with T_0 . Identify the shortest DNA sequence in test tube T_0 denote it as $O(x_\alpha, x_\beta)$.

All DNA sequences represent $O(x_\alpha, *)$ and $O(*, x_\beta)$ are extracted from test tube T_0 for all x_i where $i = 1, \dots, n$ and placed into a new test tube T_1 . Then, select the next shortest DNA sequence in test tube T_0 , denote it as $O(x'_\alpha, x'_\beta)$.

All the related DNA sequences ($O(x'_\alpha, *)$, $O(*, x'_\beta)$) are extracted from test tube T_0 and placed into another test tube named as T_2 .

Abstract $O(x'_\alpha, x'_\beta), O(x'_\alpha, x'_\beta), O(x_\alpha, x_\beta)$ and $O(x_\alpha, x'_\beta)$ from test tube T_1 and put them into a new test tube named as T'_1 .

If a shorter DNA sequence can be found in test tube T'_1 than $O(x'_\alpha, x'_\beta)$, then mix the DNA sequences together from test tube T_2 into T_1 .

The process will be repeated until all the DNA code patterns in test tube T_0 are consumptive.

However, some DNA strands in T_0 might not form a solution for a clustering group due to some errors that occurred during ligation and hybridization processes. For instance, the DNA strands may not contain DNA marking sequences as a centroid or DNA strands may not include all required patterns.

Polymerase Chain Reaction (PCR) is a process where the DNA sequences reproduce themselves to build double-stranded sequences. At the end of the process, all possible sequences were in a double stranded form.

The role of affinity-purification with a magnetic beads system process is to pick only sequences, which include all the data required for a candidate of solution in the process of procedure 3 in the proposed algorithm. In carrying out the process, the researchers incubated a double stranded DNA sequence with Watson-Crick complement of data that was conjugated to magnetic beads. The process was repeated to generate sufficient pattern sequences (the process was repeated as much as the required number of patterns obtained) to ensure all patterns were included and put them into T_1 .

Besides, only sequences that included all patterns as defined at the beginning of the process were available in T_1 . However, in the ligation and hybridization process, it might happen that some sequences may not be able to be a clustering solution (meaning, there is no marking in sequences) due to some errors. Hence, only the sequences that were marked in the test tube could be considered a solution. Another process of affinity separation by using magnetic beads was executed to ensure that only the marked sequences will undergo for the next process.

In this process, the complementarity of marking sequences was attached to magnetic beads to check the availability of marking sequences in each strand. At this point, only the sequences that contain the marking were selected and separate them into T_2 .

In order to find out the proper result in clustering, a gel electrophoresis technique was simulated to differentiate size of the sequences. The shortest DNA sequence was referred the best clustering that could be obtained from the data given. After determining the shortest DNA sequence from T_2 , another observation process was required to identify the included amount of marks as well as to determine the data point that matched to each cluster group.

5 Robustness in Clustering

In this section, the evaluation of robustness for both of the proposed method of DNA-based clustering algorithm and other conventional clustering algorithms,

will be discussed by comparing the content of evaluation approach and obtained results.

5.1 Dataset and Parameter Implementation

All the simulated data were randomly generated. Two of the three data sets were obtained from other places, while the other one was newly generated at random. As shown in Figure 2 above, dataset X1 as well as dataset X3 had a significant amount of overlap. On the other hand, dataset X2 was well separated. All data are plotted in a 2D scheme, as shown in Figure 2. This figure illustrates that noise in the data was inserted into either the *x*- or *y*-axis. Several patterns were selected for the addition of some noise, to get a new location of patterns.

Both DNA-based clustering algorithms were executed to obtain the results. These results were compared with the first results, where patterns were located at the original locations. Figure 2 shows the original location (X1-X3) and the new location (X1'-X3') obtained after the addition of noise to each pattern in the *x*- or *y*-axis.

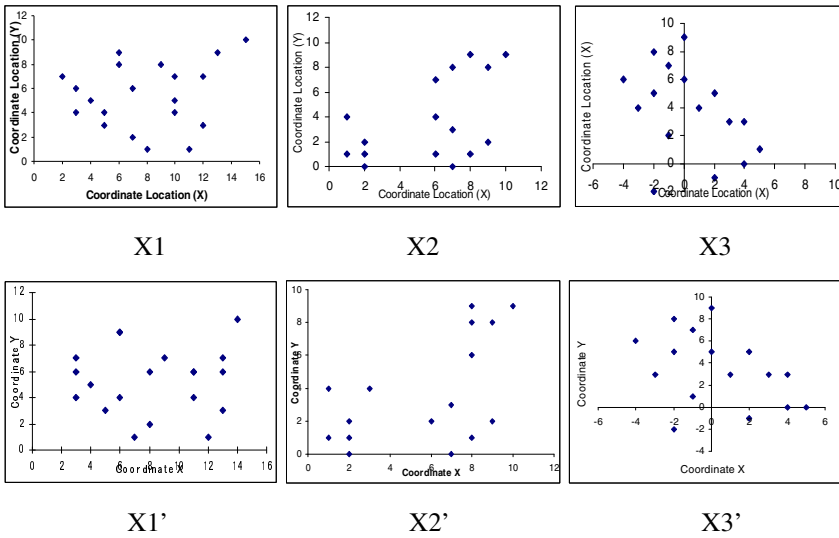


Fig 2 Data sets employed in this chapter

5.2 Robustness Evaluation of DNA-Based Clustering

DNA-based computation is a different problem-solving method, compares to silicon computer procedures. New approach relies on self annealing characters of DNA and the related DNA experiments to obtain its result. It is completely

different from other techniques such as classical calculation or computer simulation. Thus, the stability of biological experiment procedure is the most important part in ensuring DNA computation is capable of generating a robust result.

As discussed earlier, several biological and chemical factors (e.g., concentration control and temperature) might influence the stability of the biological experiment. Theoretically, the biological and chemical factors will affect results of experiments a lot, even if the same data set and algorithms were employed. However, the results should still be within almost the same range of values.

This paper examines the reliability and acceptability of two different proposed DNA-based clustering algorithms. This validity of results obtained by both techniques is scrutinized. Three different sets of data were employed to simulate the result. As stated at the beginning of the paper, two well-known conventional clustering algorithms (k-means and FCM) were also considered to show the acceptability and validity of results from these two DNA-based clustering algorithms.

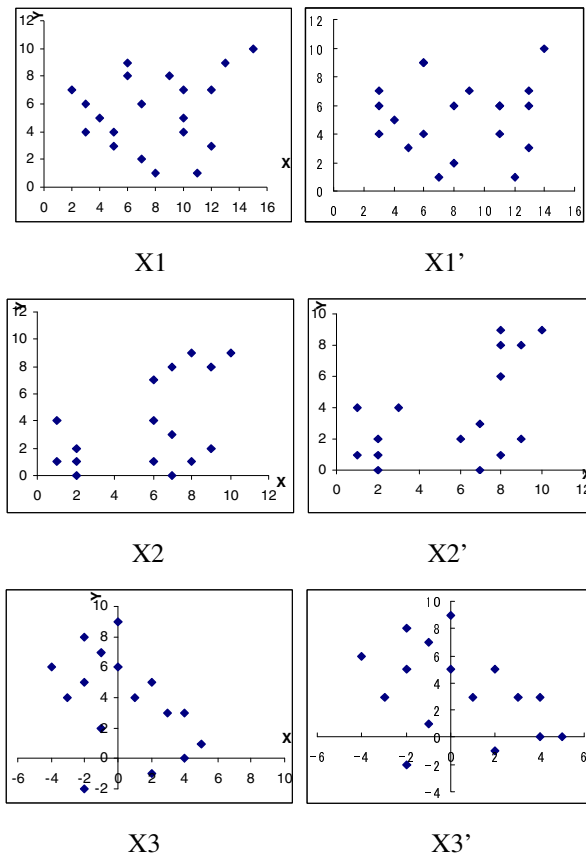


Fig. 3 Original location (X1-X3) and new location (after adding noises; X1'-X3') for sample patterns. A. Data.

To evaluate the robustness of the proposed DNA-based clustering algorithms, the following method was employed. Two different results were simulated for each data set, where the first result represented a preliminary data set and the second result was obtained from the data set after the addition of some noise. Then, the differences between these two results were observed and studied to identify the extent of which these two clustering DNA-based computation techniques could deal with some error or noise from data input. Our proposed methods were in comparison to another two conventional clustering algorithms. The patterns as shown in Figure 2 are employed to obtain the result in order to examine the robustness of proposed algorithms.

All the simulated data were randomly generated. Two of them were extracted from [2][3], while another was newly generated at random. Using all of the dataset, we changed the number of groups and the overlapping amount in such a way that:

Dataset X1 had the y- and z-axes as “non-informative”, this dataset was adopted from [2] and [3].

Dataset X2 had only the z-axes as “non informative”.

Dataset X3 had no prevalent discriminate axes.

As shown in Figure 3 above, dataset X1 had a significant amount of overlap. The same was true for dataset X3. On the other hand, dataset X2 was well separated. All data are plotted in a 2D scheme, as shown in Figure 3. This figure illustrates that noise in the data was inserted into either the x- or y-axis. Several patterns were selected for the addition of some noise, to get a new location of patterns.

Then, both DNA-based clustering algorithms were executed to obtain the results. These results were compared with the first results, where patterns were located at the original locations. Figure 2 shows the original location (X1-X3) and the new location (X1'-X3') obtained after the addition of noise to each pattern in the x- or y-axis.

6 Results and Discussion

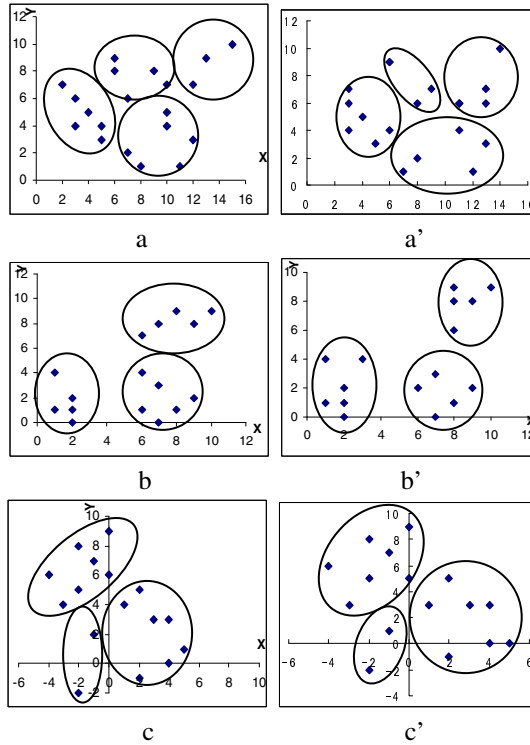
Clustering algorithms may produce different results from each other, even though the same data sets are employed. The input parameters can extremely amend the behavior and execution of the algorithm.

In addition, a clustering algorithm might produce different results when noise is added to the original data patterns. Thus, results from robust clustering algorithms should not be influenced by accumulated noise, in order to generate valid results. The aim of examining cluster validity was to evaluate the result of a clustering algorithm proposed in [2][3].

Therefore, it is important to ensure that the proposed algorithm is acceptable. Results obtained from newly generated data patterns were compared to the original results. Clustered groups were calculated based on distances between patterns

and the cluster centroid. These distances were computed in Euclidian distance for both approaches.

Figure 4 shows the results of clustered groups for data patterns before the addition of error. The approach solution cloud be viewed using a mutual distance. It is important to note that the results for samples $X1$, $X2$, and $X3$ are denoted as a , b , and c . The results for datasets $X1'$, $X2'$, and $X3'$ are denoted as a' , b' , and c' , respectively.



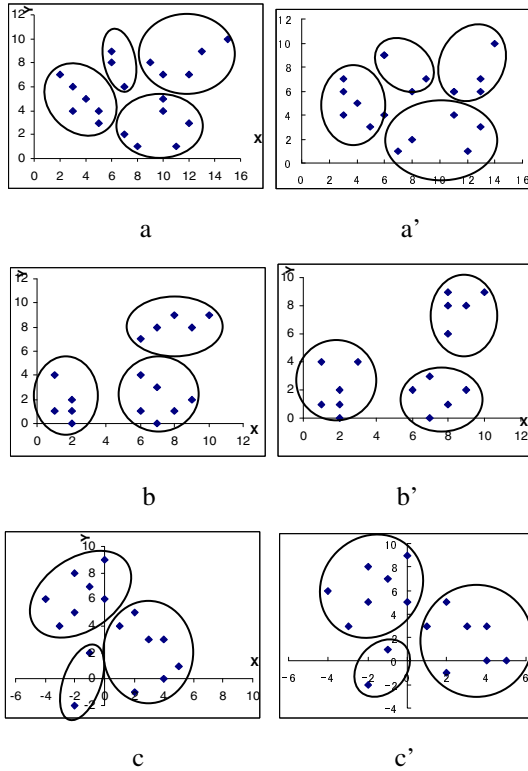
The left and right sides represents the obtained results before and after adding noises to the original data, respectively.

Fig. 4 Results of the mutual distance order approach

On the other hand, Figure 5 shows the result obtained by a proximity approach. Here the original data patterns are denoted by a , b , and c . Charts a' , b' , and c' show the result obtained from a proximity approach after the addition of some noise into the data patterns. Figures 6 and 7 show the results obtained via both conventional clustering algorithms (FMC and k-means).

DNA-based clustering showed similar results for clustered groups of patterns, both without and with the addition of noise, as shown in Figures 4 and 5. In these figures, pattern sets $X1$ and $X3$ are represented by a , a' , c , and c' , respectively. However, for sample $X2$, denoted by b and b' , a different member of each cluster

was identified (b) without noise and (b') with noise. Here c represents the result without noise and c' the result with noise. Although the pattern information changed on the y -axis, this situation did not affect the end result, as shown in Figures 4(a - a') and (b - b'), as well as in Figures 5(a - a') and 5(b - b'). However, if noise was inserted in the x -axis, as shown in Figures 4(c), 4(c'), 5(c), and 5(c'), the results were influenced in such a way that some cluster members were changed.

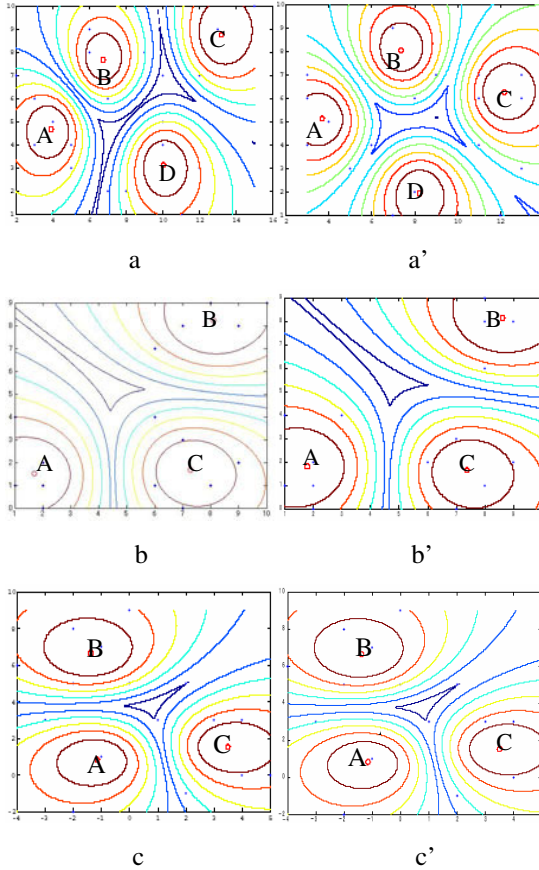


The left and right sides represents the obtained results before and after adding noises to the original data, respectively.

Fig. 5 Results of the proximity approach

Figure 6 shows the result obtained by FMC, while Figure 7 shows the result obtained by k -means. From Figures 6(a') - (c') and 7(a') - (c'), it is apparent that, for datasets that contain overlapping patterns, both of the algorithms could not accurately cluster all of the patterns into unique groups.

Some patterns could not be assigned to any specific groups or patterns that had been collected into two different groups. For example, the result for FCM in



The left and right sides represents the obtained results before and after adding noises to the original data, respectively.

Fig. 6 Result show of Fuzzy C-Means

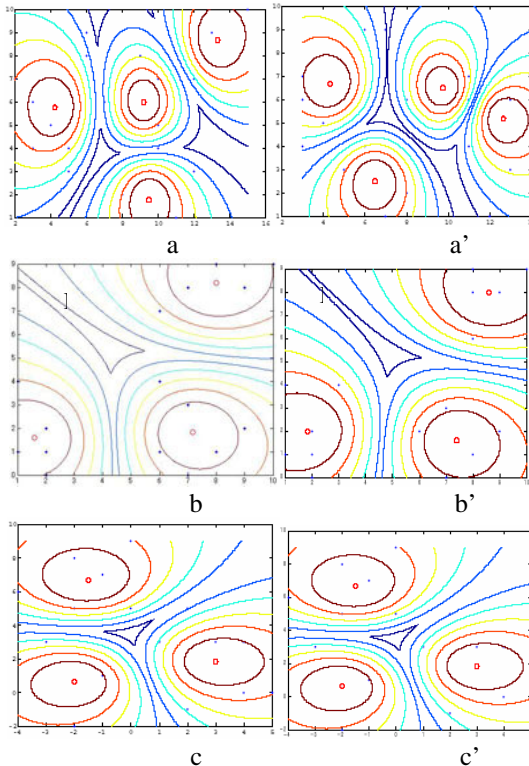
Figure 6(a), which represents $X1$, shows that one of the patterns could not be assigned to any other groups. It is the same with $X3$, where three of the patterns could not be assigned into any cluster groups, as in Figure 6(c). However, for $X2$, where the patterns were well separated, k-means was able to properly cluster them up into optimal groups.

Additionally, Figures 6(a') - (c') show the results obtained by FMC algorithms for datasets after the insertion of some noise. For $X2'$ [Figure 6(b')], where the patterns appear to be well separated from each other, FMC was able to group all patterns into unique cluster groups.

However, for datasets $X1'$ and $X3'$ [Figures 6(a') and (c')], some cluster members changed from their original settings. For dataset $X3'$, one pattern could not be

assigned to any group. For example, different numbers of cluster members were observed between datasets XI' and XI . As shown, the number of members for group C was three, when the preliminary sample was considered. This was also true for group D . However, when considering dataset XI' , the number of group C members increased to seven. Group D , however, experienced a decrease in member number from six to three.

Figure 7 illustrates the results obtained by k-means clustering algorithms. Considering the same data patterns and noise, Figures 7(a) – (c) demonstrate the results before noise insertion. Figures 7(a') - (c') present the results obtained after noise insertion. From Figures 7(a) - (c), it is apparent that k-means was capable of clustering data properly only if the patterns were well separated, as shown in Figure 7(b). However, compared to FMC, k-means was also able to cluster data perfectly, even when the patterns had some overlapping data, as shown in Figures 7(a) and (c).



The left and right sides represents the obtained results before and after adding noises to the data, respectively.

Fig. 7 Results of K-Means

In this study, after the insertion of noise, patterns were assigned into two different groups (groups *B* and *C*). This was also true for dataset *X3*, where k-means was unable to assign one of the patterns to any cluster group. Compared to DNA-based clustering, FMC and k-means did not select one of the patterns as a centroid for each cluster group. However, the centroid was identified by the central point of the group. In this situation, if attempting to solve a practical real-life problem, such as a distribution center problem where it is necessary to select a city from the list as a centroid, it is essential to not point out a place as the centroid for the cluster group.

Other than this, the proposed algorithms for DNA-based clustering were capable of grouping all patterns in both circumstances, with or without noise, into unique cluster groups. In conventional clustering methods, a few patterns are sometimes unable to be grouped into an exclusive cluster group, owing to their limitations. Hence, these ungrouped patterns should undergo some cleaning or normalization process in order to adequately group them. Nonetheless, some other techniques, such as standard deviations or geometric means, can be employed to solve the problem of these ungrouped data in order to improve the clustering results.

7 Conclusions

This paper provided detailed information on a new clustering algorithm using DNA computing. There are two main issues were discussed in this paper; (i) evaluation of robustness by considering noise input data; and (ii) evaluation of performance and validity of obtained results from statistical point of view. In order to achieve this objective, three different conventional clustering algorithms and two validity indices are considered in this study were compared with proposed DNA-based clustering algorithms.

This paper also explained the procedures of employing the bio-chemical technique in solving the clustering problem. The experimental results were presented to demonstrate the feasibility of the approach in determining the node that should become the centroid for each clustering group without considering any prior information regarding the numbers of clusters provided at the beginning of the process.

This occurred due to the process of ligation and hybridization with Watson-Crick complementarily that are able to produce all the feasible end solutions in the parallel way. Through all these possibilities, all the possible numbers of clustering groups are generated. Then, the gel electrophoresis process will sort these results based on the length of the DNA strand itself. The shortest strand should become the best solution. However, if the user has their own criteria, the user can choose their own best results based on the length of the strands.

One of the issues that should be considered in designing a clustering algorithm is the robustness of the algorithm itself. In this study, all DNA sequences that represent patterns in the final strand are calculated and scrutinized. Of course, if only the small number of patterns is considered, the result might be eventually affected if any inappropriate sequences interfere during the process.

Though DNA might be time-consuming, however, DNA needs only single time process to produce all the feasible solutions through the ligation and hybridization process without undergoing any repetition process. When considering a large number of data, DNA computing approach is very effective in comparison to other techniques. However, in dealing with small-medium data, DNA computing might not be the best method to be employed. The reason is that DNA computing requires huge processing time as in comparison to other techniques. Hence, it is inappropriate for small or medium size of data.

In this work, we studied the influence of some error or noise on patterns obtained by a DNA computing approach in clustering. Two DNA computing algorithms, the mutual distance approach [2] and proximity approach [3], were employed as case studies.

Based on the findings from both of these approaches, it has been confirmed that noise occurring on the y -axis does not affect the clustering result. Nevertheless, some changes in information for patterns on the x -axis will influence the end clustering result. Because any pattern changes on the x -axis will cause significant changes in the pattern position. On the other hand, changes on the y -axis do not result in such obvious changes in pattern position.

Therefore, it can be concluded that a small change in pattern information will not affect the end results. However, if pattern information is significantly amended, changes in cluster members will be resulted in. Employing DNA procedures is highly sensitive to any change in patterns. However, DNA computing can be proposed for reliable clustering algorithms, the results through this proposed techniques are acceptable even noise existing.

The robustness of clustering results is an important concern in the context of clustering research. A valid result enables the employment of the proposed technique in solving other application problems. Based on this study proved that the proposed algorithm is sufficiently robust in dealing with some unexpected errors, mainly concerning the data input. There are different points of view regarding reliability in the research area of DNA computing, compared to the clustering research field. However, this paper aimed to focus on validating clustering results through the DNA computing procedure, to prove the robustness of clustering algorithms when dealing with some error or noise in the data.

However, if the sufficiently large number of patterns is considered, the small number of inappropriate sequences such as outliers is not affecting the results of clustering. It is important to note that, that instead of distances themselves the study also considered ordering, which is far more robust than numeric values of distances.

References

- [1] Adleman, L.M.: Molecular Computation of Solutions to Combinatorial Problems. *Science* 266(11), 1021–1024 (1994)
- [2] Bakar, R.B.A., Watada, J., Pedryzc, W.: DNA approach to solve clustering problem based on a mutual distance order. *Biosystems* 91(1), 1–12 (2008)

- [3] Bakar, R.B.A., Watada, J.: A proximity approach to DNA based clustering analysis. *International Journal of Innovative Computing, Information and Control (IJICIC)* 4(5), 1203–1212 (2008)
- [4] Oehler, K.L., Gray, R.M.: Combining image compression and classification using vector quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17(5), 61–473 (1995)
- [5] Shopbell, P.L., Britton, M.C., Ebert, R.: Making the most of missing values: object clustering with partial data in astronomy, astronomical data analysis software and system XIV. *ASP Conference Series*, vol. 30 (2005)
- [6] Jiang, T., Tuzhillin, A.: Segmenting customers from population to individuals: Does 1-to-1 keep your customer forever. *IEEE Transaction on Knowledge and Data Engineering* 18(10), 1297–1311 (2006)
- [7] Jimmy, L., Karakos, D., Fushman, D.D., Khudanpur, S.: Generative content models for structural analysis of medical abstracts. In: *Proceedings of the 2006 Workshop on Biomedical Natural Language Processing (BioNLP 2006)*, New York City (June 2006)
- [8] Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. *ACM Computer Surveys* 31(3) (September 1999)
- [9] Pedrycz, W.: *Knowledge-based clustering: From data to information granules*. Wiley Interscience, Hoboken (2005)
- [10] Franti, P., Xu, M., Karkkainen, I.: Classification of binary vectors by using Δ SC distance to minimize stochastic complexity. *Journal of Pattern Recognition* 24, 65–73 (2003)
- [11] Lu, X.-g., Lin, et al.: Gene cluster algorithm based on most similarity tree. In: *Proceedings of the Eighth International Conference on High-Performance Computing in Asia-Pacific Region (HPCASIA 2005)*, Beijing, November 30–December 3 (2005)
- [12] Cleju, I., Franti, P., Wu, X.: Clustering Based on Principal Curve. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) *SCIA 2005*. LNCS, vol. 3540, pp. 872–881. Springer, Heidelberg (2005)
- [13] Jain, A.K., Law, M.H.C.: Data clustering: A user's dilemma. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) *PRMI 2005*. LNCS, vol. 3776, pp. 1–10. Springer, Heidelberg (2005)
- [14] Volfovsky, N., et al.: A clustering method for repeat analysis in DNA sequences, *Genome Biology Publication*, Citing Internet sources (2001), <http://genomebiology.com/2001/2/8/research/0027>
- [15] FitzGerald, P.C., Shlyakhtenko, A., Mir, A.A., Vinson, C.: Clustering of DNA sequences in human promoters. Cold Spring Harbor Laboratory Press (2004); ISBN 1088-9051/04, <http://www.genome.org>
- [16] Sang, L., et al.: CLAGen: A tool for clustering and annotating gene sequences using a suffix tree algorithm. *BioSystems* 84, 175–182 (2006)
- [17] Joseph, Z.B., Gifford, D.K., Jaakkola, T.S.: Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics* 17(suppl.1), S22–S29 (2001)
- [18] Kim, S.Y., Lee, W.L., Bae, J.S.: Effect of data normalization on fuzzy clustering of DNA microarray data. *BMC Bioinformatics* 7,134 (2006), <http://www.biomedcentral.com/1471-2105/7/135>

Advances in Automated Neonatal Seizure Detection

Eoin M. Thomas, Andrey Temko, Gordon Lightbody, William P. Marnane, and Geraldine B. Boylan

Abstract. This chapter highlights the current approaches in automated neonatal seizure detection and in particular focuses on classifier based methods. Automated detection of neonatal seizures has the potential to greatly improve the outcome of patients in the neonatal intensive care unit. The electroencephalogram (EEG) is the only signal on which 100% of electrographic seizures are visible and thus is considered the gold standard for neonatal seizure detection. Although a number of methods and algorithms have been proposed previously to automatically detect neonatal seizures, to date their transition to clinical use has been limited due to poor performances mainly attributed to large inter and intra-patient variability of seizure patterns and the presence of artifacts. Here, a novel detector is proposed based on time-domain, frequency-domain and information theory analysis of the signal combined with pattern recognition using machine learning principles. The proposed methodology is based on a classifier with a large and diverse feature set and includes a post-processing stage to incorporate contextual information of the signal. It is shown that this methodology achieves high classification accuracy for both classifiers and allows for the use of soft decisions, such as the probability of seizure over time, to be displayed.

Keywords: Neonatal EEG analysis, biomedical signal classification.

1 Introduction

Neonatal seizures are estimated to occur in 1-5% of babies [1] and can represent an important sign of neurological dysfunction. The main cause of neonatal seizures is asphyxia [2], with low birth weight and premature babies particularly at risk [3].

Eoin M. Thomas · Andrey Temko · Gordon Lightbody · William P. Marnane
Department of Electrical and Electronic Engineering, University College Cork, Ireland

Geraldine B. Boylan
School of Medicine, University College Cork, Ireland

Clinical signs may be absent in as many as 85% of neonatal seizures, thus requiring for the electroencephalogram (EEG) in order to detect all seizures [4]. This has led to the EEG becoming the gold standard for neonatal seizure detection [5]. Neonatal EEGs are interpreted by neurophysiologists who are generally not available in the neonatal intensive care unit (NICU) on a 24 hour basis. Therefore, the main aim of an automated neonatal seizure detection system is to assist clinical staff in a NICU in interpreting the EEG. An automated seizure detector would also prove useful in highlighting areas of interest for review by a clinical neurophysiologist, in order to reduce the work-load of the clinician.

The background patterns of normal, full term babies are in the delta (0-4Hz), theta (4-7Hz), alpha (7-12Hz) and beta (12-30Hz) bands. Patterns range from continuous delta and theta activity when the child is awake to *tracée alternant* (bursts of delta, theta, alpha and beta activity alternating with theta and alpha activity) patterns during quiet sleep [6]. In sick patients, such as patients with seizures, the background activity is abnormal. For instance, in severely sick patients the background EEG may be continuously low amplitude ($<30\mu V$) or exhibit burst-suppression patterns. Furthermore, medication such as anti-epileptic drugs may be a confounding factor in the interpretation of the EEG [6]. An example of background EEG in a sick patient is shown in Figure 1.

Neonatal seizures in EEG are defined as periods of increased periodicity for a duration of over 10 seconds [7]. These events can be localized to a single channel but are more frequently observed on a number of channels simultaneously. Kitayama *et al.* [8] reported that the frequency range of neonatal seizures is between 0.5-13Hz with the dominant components in the 0.5-6Hz range, based on wavelet analysis. An example of seizure EEG is shown in Figure 2.

The EEG is also subject to artifacts, both physiological and environmental. Artifacts occur due to movement of the patient, handling of the patient, mains noise contamination, electrode detachment and respiration among others. These events are problematic for seizure detection as an artifact may mask a seizure trace, or



Fig. 1 Background EEG in a sick patient. In this example, the EEG shows primarily delta and theta activity

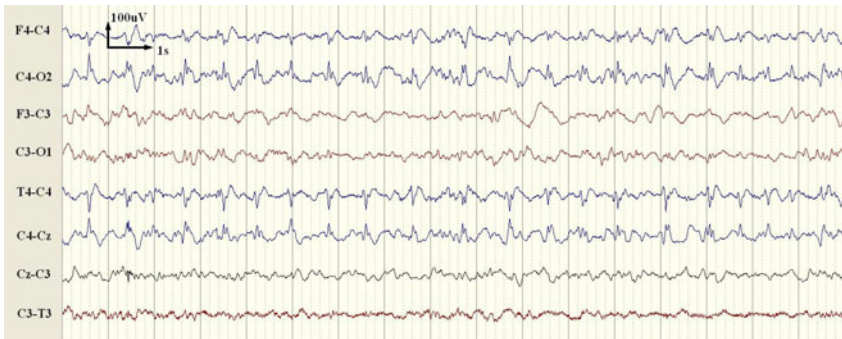


Fig. 2 Example of seizure EEG. Here, a repetitive spike pattern can be seen on channels F4-C4, C4-O2, T4-C4 and C4-Cz.

conversely, an artifact may appear as a repetitive trace and be misinterpreted by the automated detector as a seizure. For a human expert, it may be possible to discern artifacts using additional recordings such as video, the electrocardiogram and a respiration trace.

The majority of approaches to neonatal seizure detection can be grouped into two classes: threshold based methods and classifier based methods. Threshold based methods consist in analysing the EEG using a small number of descriptors from which a decision is made using empirically derived thresholds. Recent examples of this methodology include the work of Navakatikyan *et al.* [9] and Deburchgraeve *et al.* [10]. Navakatikyan *et al.* generated features from the peaks and troughs of successive waves and additionally extracted a correlation coefficient between successive waves. A threshold based decision making routine was then used to generate initial decisions from these features, followed by a postprocessing scheme.

Deburchgraeve *et al.* [10] proposed a system composed of two independent routines. One routine analyses the “spikiness” of the EEG, while the second analyses the EEG for repetitive activity. A drawback of threshold based approaches is that controlling the trade-off between good detections and false detections is complicated by the multiple thresholds involved in the decision stage.

Classifier based methods employ elements of pattern recognition to classify a set of features using a data-driven decision rule. Aarabi *et al.* used feature selection methods to select an optimal subset of features for use in an Artificial Neural Network (ANN) [11] and, more recently, trained an ANN to classify neonatal EEG into several background states and two seizure states [12]. Greene *et al.* [13] investigated linear, quadratic and regularised discriminant analysis for neonatal seizure detection. Recently, Mitra *et al.* [14] used neural networks as part of a neonatal seizure detector; however, a large number of heuristic rules and thresholds are also employed making it unclear which aspects of the detector contributed towards the final decision.

Classifier based methods have several advantages over threshold methods. The trade-off between good detections and false detections can be controlled via a single

parameter. This allows for the system to be tuned to a particular application. For instance, a system used for reviewing records may be set to obtain a higher rate of good and false detections than a system used for real time monitoring, due to false detections being less critical during review. Furthermore, novel features can be easier to incorporate into a classifier based system than into a threshold based system.

Support vector machines (SVMs) and Gaussian mixture models (GMMs) are presented here as examples of discriminative and generative approaches to classification. Recent work on statistical machine learning has shown the advantages of discriminative classifiers such as SVMs [15] in a range of applications. Examples of seizure detectors based on SVMs can be found in the field of epileptic seizure detection in adults such as [16, 17, 18]. An SVM was shown to produce good results in a patient dependent neonatal seizure detection system [19], however, this study was limited to only one patient.

Classifiers based on GMMs have been validated in fields such as speech recognition [20] and have been employed to classify EEG in biomedical applications such as brain computer interfaces [21] and person authentication [22]. A seizure detector based on GMMs was proposed by Meng *et al.* [23] to accurately classify intracranial recordings from adult patients.

2 Data and Experiment Setup

The dataset employed in this study was comprised of recordings from 17 fullterm neonates (gestational age range 39-42 weeks) produced in Cork University Maternity Hospital, Cork, Ireland. A Viasys NicOne video EEG machine was used to record multichannel EEG at 256Hz using the 10-20 system of electrode placement modified for neonates. In this study, 8 bipolar EEG channels are used (F4-C4, F3-C3, T4-C4, C4-CZ, CZ-C3, C3-T3, T4-O2, T3-O2). The dataset contained over 267 hours of EEG. A total of 705 seizure events were annotated by 2 experienced neonatal electroencephalographers, further information on the dataset can be found in Table 1.

The nature of problem is such that patient-specific data is not available to the detector prior to testing. Thus, it is important to retain this characteristic in the experimental setup. Furthermore, due to the limited availability of neonatal EEG data, it is important to maximise the use of data. For these reasons, the performance of the detector is assessed by patient-independent leave-one-out (LOO) cross-validation. The training dataset is obtained from 16 patients and the remaining patient constitutes the testing dataset, this process is then repeated until each patient has been used as the test subject. The mean result is then reported, this can be viewed as an appropriate estimate of the performance of the system on unseen patients, as LOO is known to have low bias despite high variance [24].

The training data used for each classifier is determined by two factors: availability and processing time. For the seizure class, per channel annotations are required to indicate which channels the seizure event occurs on and thus which channels should

Table 1 EEG dataset for the cross-validation set

Patient	Record length (hours)	Seizure events	Mean seizure duration (minutes)
1	18.23	17	1.50
2	24.74	3	6.17
3	24.24	149	2.29
4	26.10	60	1.05
5	24	49	5.90
6	5.69	41	1.16
7	24.04	6	1.07
8	24.53	17	5.95
9	24.04	156	5.27
10	10.06	25	5.44
11	6.19	15	5.44
12	12	29	2.18
13	12.13	25	4.10
14	5.48	11	8.57
15	12.16	59	2.08
16	7.63	31	10.39
17	6.64	12	8.54
Total	267.90	705	-

be used in training. However, neonatal EEG is not typically annotated in such detail as it is time consuming for the clinician. In this study, 2 minutes of seizure data are annotated per channel for each patient, this equates to ~ 1500 data points during training. Thus, the seizure class training data is identical for both classifiers.

For the non-seizure class, per channel annotations are not required as a non-seizure annotation implies that all channels are non-seizure. Thus the number of non-seizure training exemplars is limited only by the training time. For SVM, the training time rises exponentially with the number of data. Furthermore, SVMs are known to perform well with sparse datasets, thus a high number of training data are not required [15]. Here, $\sim 10^4$ points are randomly selected from all training patients to make up the training dataset. In contrast, the training time for GMMs increases more linearly with data. Furthermore, better results are obtained when the PDF of the training data approaches the true PDF of the data as the GMM is a density estimator. Thus, increasing the size of the training dataset results in a sample PDF which better approximates the true PDF of the data. Here, $\sim 10^6$ non-seizure datapoints are used to train the non-seizure GMM.

3 Probabilistic Classification Framework

The system is designed using a late integration architecture, where each EEG channel is processed and classified independently prior to combination of the

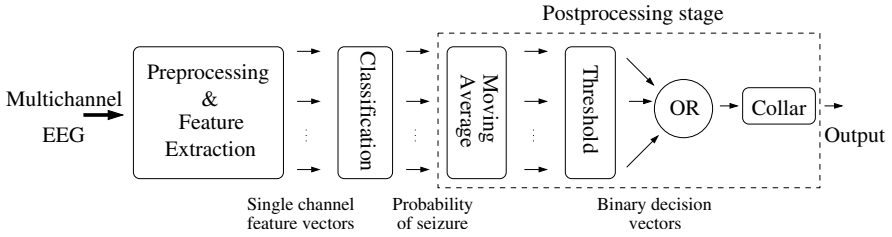


Fig. 3 Neonatal seizure detector system diagram. The Classification block is explained in further detail in section 3.2.

decisions as seen in Fig. 3. The following section gives an overview of each stage of the detector.

3.1 Preprocessing and Feature Extraction

In order to reduce the computational time and memory load of the feature extraction stage, the EEG is downsampled to 32Hz with an anti-aliasing filter set to 12.8Hz. The EEG is then segmented into epochs using an 8s sliding window with 50% overlap.

The features were primarily obtained or modified from a feature comparison study by Greene *et al.* [25]. A total of 55 features are extracted from each epoch and are briefly described here under the headings of frequency domain, time domain and information theory techniques.

Frequency-domain features: A number of features are extracted from the power spectral density (PSD) of the epochs, and are thus considered frequency-domain features. The (PSD) of each epoch is obtained using a 256 point fast Fourier transform (FFT).

The power in the PSD from 0Hz to 12Hz is extracted to quantify increases in the power of the EEG, this is referred to as the total power. Analysis of the power in specific subbands has been validated in adult studies [26, 16]. Here, the power in bands with a width of 2Hz is extracted from the PSD with an overlap of 1Hz between bands, e.g. 0-2Hz, 1-3Hz, etc. In addition, the power bands are normalised by the total power in order to minimise the effect of high power artefacts. Both normalised power bands and non-normalised power bands are used as features. The dominant-peak frequency as proposed by Gotman *et al.* [27] is extracted from the PSD.

Three features are extracted based on spectral edge frequency (SEF), defined as the frequency under which a certain percentage of the power in the PSD lies. The three features are calculated corresponding to 80%, 90% and 95% of the power in the PSD. The entropy of the PSD, referred to as spectral entropy, is used to give measure of the complexity of the PSD, using the same method as Greene *et al.* [25]. A final frequency domain feature is calculated not from the FFT but rather based on wavelet decomposition. The use of wavelet analysis has been validated by [8]. The

Daubechey 4 wavelet is used to decompose the EEG into 8 coefficients, the energy in the 5th coefficient corresponding to 1-2Hz is used here as a feature.

Time-domain features: Aarabi *et al.* [11] performed relevance and redundancy testing of a large set of simple features. Many of these features came from simple statistical analysis of the epoch or the first and second derivative of the epoch. Here, a number of features selected during relevance and redundancy testing are used. These include the root mean squared amplitude along with the skewness and kurtosis of the epoch. The variances of the first and second derivative of the epoch are calculated. The number of zero crossings is calculated from the epoch and from the first and second derivative of the epoch.

Additionally, a number of time domain features were implemented based on the feature comparison work of Greene *et al.* [25]. The number of maxima and minima from each epoch. Hjorth [28] designed 3 parameters, activity, mobility and complexity, for EEG analysis which are used as features here. Activity is simply the variance of the signal. Mobility is the standard deviation of the first derivative of the signal divided by standard deviation of the signal. Complexity is the standard deviation of the second derivative of the signal divided by the standard deviation of the first derivative, all of which is divided by the mobility of the signal.

Features used for epileptic event detection in adults are also employed such as nonlinear energy and curve length. Nonlinear energy was used by D'Alessandro *et al.* [29] for seizure prediction in epileptic patients. Curve length, or line length, was proposed by Esteller *et al.* [30] as an indicator of seizure onset. The final time domain features extracted from the EEG are based on autoregressive modelling. An autoregressive model is fitted to the first half of the epoch, the residual prediction error is then obtained from the second half of the epoch and used as a feature. This technique is used to obtain nine features by using nine different model orders (from 1 to 9).

Information theory features: In a study by Faul *et al.* [31], results of ANOVA tests found that information based features showed high separation between seizure and non-seizure EEG. In this study, a number of features quantifying the complexity of the EEG and showing highest separation in Faul *et al.* [31] are used. Shannon entropy is computed from a histogram of the EEG epoch. Singular value decomposition (SVD) entropy is extracted from the EEG For this feature, the EEG is first embedded using a time delay of 1 and an embedding dimension of 20. The singular value decomposition algorithm is applied to the embedded matrix yielding a set of singular values. The entropy of the normalised singular values is then calculated as a measure of the complexity of the EEG. Fisher information is also used as a feature based on the singular values of the EEG, but this feature is less influenced by the power in the signal than SVD entropy.

3.2 Classification

Two classifiers are compared for the decision making task: an SVM classifier and a GMM classifier. The SVM is a type of discriminative classifier which maximises

the distance from the decision boundary to the nearest datapoints (called support vectors) in the training data. In contrast, the GMM is a type of generative classifier which models the probability density function of a random variable as a weighted sum of Gaussian distributions.

The SVM Classifier

Support vector machines are part of a family of discriminative classifiers known as maximum margin classifiers. For the most basic case of classifying two linearly separable sets from data $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathfrak{R}^d$ with corresponding labels $Y = \{y_1, \dots, y_n\}$, $y_i \in \{1, -1\}$ the problem may be stated as finding the hyperplane parameters $\boldsymbol{\omega}$ and b satisfying:

$$y_i(\boldsymbol{\omega}\mathbf{x}_i + b) \geq 1, \forall i. \quad (1)$$

A number of hyperplanes may satisfy this equation, however in order to obtain a classifier with good generalization characteristics the margin between the hyperplane and the nearest points of each class, known as support vectors, should be maximal. The margin is given by $\frac{2}{\|\boldsymbol{\omega}\|}$, thus the problem of finding the optimal hyperplane can be stated as follows:

$$\text{minimise } \frac{1}{2}\|\boldsymbol{\omega}\|^2, \quad \text{subject to } y_i(\boldsymbol{\omega}\mathbf{x}_i + b) \geq 1, \forall i. \quad (2)$$

Data observed in real conditions is frequently inseparable. To account for this, the decision boundaries can be softened by introducing a *slack* positive variable ξ_i . However, to avoid the trivial solutions caused by a large ξ_i , we introduce a penalization cost in the objective function. The problem can thus be formulated as:

$$\text{minimise } \frac{1}{2}\|\boldsymbol{\omega}\|^2 + C \sum_{i=1}^n \xi_i, \quad \text{subject to } y_i(\boldsymbol{\omega}\mathbf{x}_i + b) \geq 1 - \xi_i, \forall i, \quad (3)$$

where C is a positive regularization constant which controls the degree of penalization of the slack variables. For a non-linearly separable classification problem the data must first be mapped onto a higher dimensional feature space where the data are linearly separable. This is achieved via a kernel trick. The kernel can be thought of as a non-linear similarity measure between two datapoints, e.g. support vector \mathbf{x}_i and point \mathbf{x}_j . The kernel function used here is the radial basis function (RBF):

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-|\mathbf{x}_i - \mathbf{x}_j|^2 / 2\sigma} \quad (4)$$

The Gaussian kernel parameter σ from (4) and generalisation parameter C from (3) are optimised over five-fold cross validation of the training dataset. The model used in testing is then obtained over the full training dataset using this pair of parameters.

The output of the SVM is bounded between $[0 \ 1]$ via a sigmoid function whose parameters are estimated on the training dataset as described in [32]. This is

implemented both to produce an estimate of probability of seizure for the SVM and to improve the final results. For unbalanced problems, such as seizure detection, decisions made with a threshold given by the sigmoid function were shown to be significantly better than those obtained with the original threshold of zero applied to the distance to the separating hyperplane [32]. Additionally, the conversion to probabilistic values facilitates the choice the desired operating point as the threshold has to be chosen from the bounded interval [0 1]. A block diagram of the SVM classification stage is given in 4.

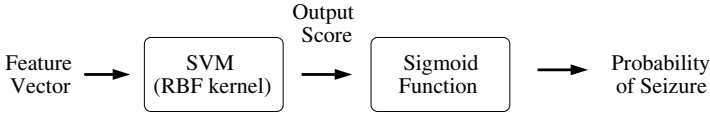


Fig. 4 SVM classifier with probabilistic postprocessing

The GMM Classifier

The GMM classifier is a generative classifier, that is the classifier models the underlying probability density function (PDF) of each class. For this reason a GMM is used per class to obtain the likelihood of class membership. The likelihoods are then combined to create a probability of class membership based on Bayes' theorem.

A GMM represents the PDF of a random variable, $\mathbf{x} \in \mathcal{R}^d$, as a weighted sum of k Gaussian distributions:

$$p(\mathbf{x}|\theta) = \sum_{m=1}^k \alpha_m p(\mathbf{x}|\theta_m), \text{ where } \sum_{m=1}^k \alpha_m = 1, \text{ and } \alpha_m > 0, \forall m. \quad (5)$$

Here θ is the mixture model, α_m corresponds to the weight of component m and the density of each component is given by the normal probability distribution:

$$p(\mathbf{x}|\theta_m) = \frac{|\Sigma_m|^{-1/2}}{(2\pi)^{d/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_m)^T \Sigma_m^{-1} (\mathbf{x} - \boldsymbol{\mu}_m) \right\}. \quad (6)$$

During training, the parameters α , $\boldsymbol{\mu}$ and Σ are optimised iteratively via the Expectation Maximisation algorithm [33] in order to maximise the log-likelihood of the model. Given a group of n independent and identically distributed samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, the log-likelihood corresponding to a mixture model θ is given by

$$L(\mathbf{X}; \theta) = \log \prod_{i=1}^n p(\mathbf{x}_i; \theta) = \sum_{i=1}^n \log \sum_{m=1}^k \alpha_m p(\mathbf{x}_i; \theta_m). \quad (7)$$

In the testing stage, a likelihood estimate is obtained for the seizure class, defined by the model θ_s , and for the non-seizure class, defined by the model θ_n , as shown

in Figure 5. The likelihood estimates are then combined to yield the posterior probability of seizure for the sample using Bayes' theorem [34].

To account for the lack of discriminative information in the training of the GMM, linear discriminant analysis (LDA) is used to preprocess the feature vectors. This transform maximises the ratio of between class covariance to within class covariance of the training data, yielding a set of discriminant vectors [35]. The set of discriminant vectors is used to create a matrix transform in order to project the original feature vectors into a new feature space. Directions of low discrimination are subsequently removed from the LDA matrix, allowing for a reduction in the dimensionality of the projected feature space, while retaining the discriminating ability of the original space. It was found that a final feature dimension of 30 yielded best results [36]. This transformation matrix is obtained over the training data and is then used for the testing data. Here, a GMM with 8 full covariance Gaussian distributions is used as in [36].

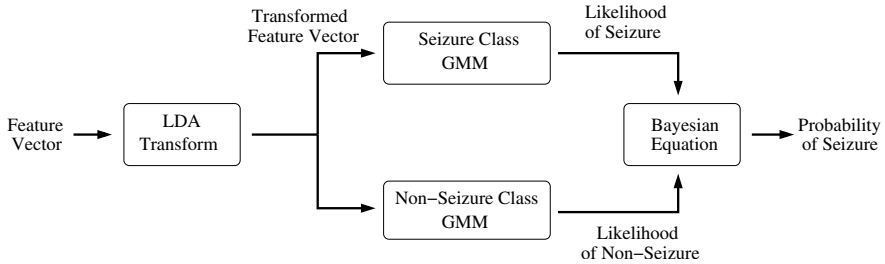


Fig. 5 GMM classifier with LDA preprocessing

3.3 Postprocessing

The probability of seizure obtained from the classification stage is postprocessed to include contextual information. The postprocessing stage is highlighted in Fig. 3 as it consists of multiple operations. First, the per channel probabilities of seizure are filtered using a 15-epoch central moving average filter. This is performed in order to smooth the probability of seizure vector and serves mainly to remove short false positives. A seizure is declared on a channel if the filtered probability of seizure exceeds a certain threshold, resulting in a binary decision for each epoch. The per channel decisions are fused into a single decision vector where a seizure is declared if found on any channel. Finally, the collar technique used in speech processing applications is applied here. Seizure decisions are extended, or grown, from either side by 40s to compensate for possible difficulties in detecting the start and end of seizure events. The collar operation is useful in combination with the moving average filter as the moving average filter may reduce the duration of detections by smoothing the sharp onset and offset of events. An example of the effects of postprocessing is shown for a single channel of EEG in Fig. 6.

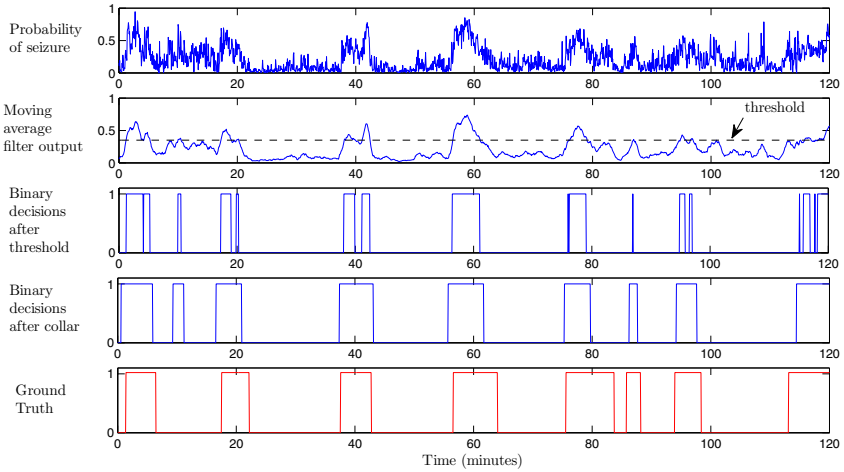


Fig. 6 Example of postprocessing on a single channel of EEG. The original probability of seizure is shown first. This probability of seizure is then filtered using a 15 point central moving average filter. A threshold is applied to obtain binary decisions. The binary decisions are then expanded by 40s using the collar operation. The last figure shows the ground truth. For this section all events are detected with one false detection at 10 minutes.

3.4 Metrics

The final output of the detector is a binary decision. In this decision, a 1 is referred to as a positive decision and indicates seizure. A 0 is referred to as a negative decision and corresponds to the non-seizure class. This binary decision vector is compared to the ground truth (annotations from the electroencephalographer) in order to produce a number of metrics.

Epoch Based Metrics

Epoch based metrics are obtained from the confusion matrix shown in Fig. 7, which shows the number of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) epochs.

Sensitivity is the percentage of seizure epochs correctly classified. Specificity is the percentage of non-seizure epochs correctly classified. The sensitivity of the detector can be increased by reducing the specificity of the system via a decision threshold. A plot of sensitivity against specificity over the entire range of decision thresholds is known as the Receiver Operating Characteristic (ROC) curve. The area under the ROC curve is reported as the ROC area. Precision is the percentage of correct positive decisions. Precision is plotted against the recall of the system to obtain a Precision Recall (PR) curve, where recall is equivalent to sensitivity. The

		Ground Truth	
		1	0
Binary decision	1	TP	FP
	0	FN	TN

Fig. 7 Confusion matrix

PR curve provides added information about the performance of the system, as it is affected by unbalanced prior distributions of classes.

$$\text{Sensitivity (Recall)} = \frac{TP}{TP + FN} \times 100$$

$$\text{Specificity} = \frac{TN}{TN + FP} \times 100$$

$$\text{Precision} = \frac{TP}{TP + FP} \times 100$$

Event Based Metrics

Event based metrics are given for the “any overlap” grading scheme. If any positive decision correctly overlaps with an annotated seizure event, the entire event is considered detected. In a continuous train of positive decisions, false positives do not count as false detections provided there is any overlap between any positive decision and an annotated seizure event. Succeeding false positive decisions are grouped as a single false detection event. The Good Detection Rate (GDR) is defined as the percentage of seizure events correctly detected. False detections are reported using the False Detections per hour (FD/h) metric. The any overlap grading scheme can result in misleading results if the false detections are of an extended duration. Thus, the Mean False Detection Duration (MFDD) is also reported.

4 Results

4.1 Results Over All Patients

The mean ROC curves over all patients are plotted in Fig. 8, from which it can be seen that the mean ROC area for the SVM system (96.3%) is marginally higher than that of the GMM system (95.8%). However, both systems produce high values of sensitivity and specificity over all patients. The difference between the systems

is more evident in the mean PR curves given in Fig. 9. The precision of the SVM system is higher than that of the GMM system for low values of sensitivity. This behaviour indicates that for low percentages of detected seizure epochs, a high percentage of positive decisions are correct, especially for the SVM. In the desired sensitivity range of 0.7-1, however, the precision of the GMM system is comparable to that of the SVM system.

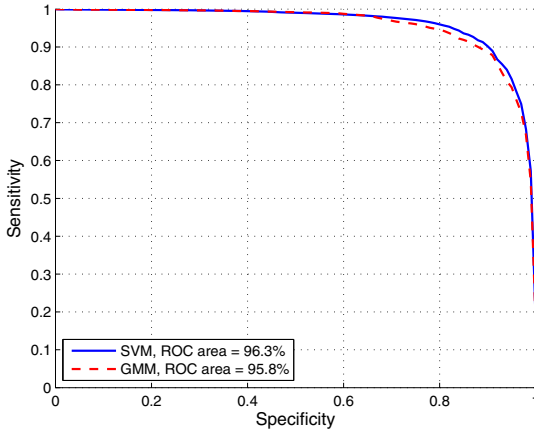


Fig. 8 ROC curve. The SVM ROC area is larger than that of GMM.

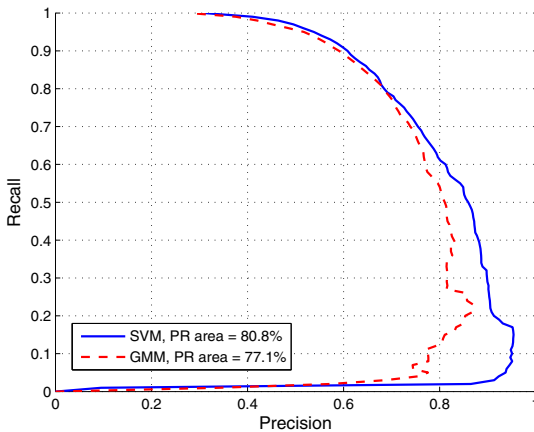


Fig. 9 PR curve. This curve shows that the SVM precision is higher than the GMM precision, particularly for lower values of recall.

The event based metrics are presented in Fig. 10, in which the GDR is plotted against the number of FD/h. For low FD/h, both classifiers achieve a GDR above 50%. Overall, it can be seen that the SVM system detects a higher percentage of seizures than the GMM system for the same FD/h. The GMM system has a slightly lower MFDD for 0.25 and 0.5 FD/h.

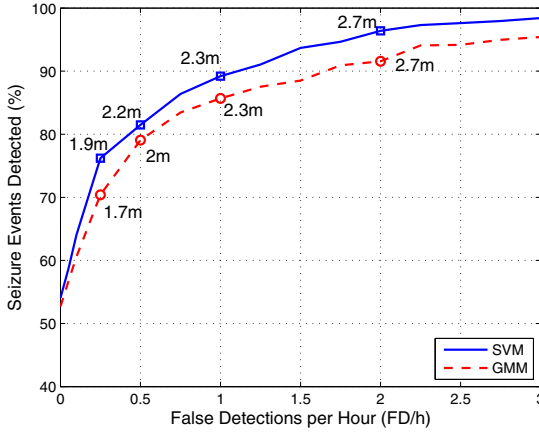


Fig. 10 GDR as a function of FD/h for the SVM and GMM systems. The mean false detection duration is shown for 0.25, 0.5, 1 and 2 FD/h in minutes.

In order to gain further insight into the performance of the systems, false detections and missed seizures were analysed with both systems operating at 0.5 FD/h. Seizure events were grouped into 4 subsets according to the duration of the seizure event. The percentage of seizures detected is shown for each subset in Fig. 11. There is a clear trend indicating that the detection rate is linked to the duration of the seizures (t_s). In particular, seizures of duration less than one minute achieve less than 60% good detection, whereas seizures lasting over 2 minutes achieve over 94% good detection. It can be seen that the largest difference between the classifiers is for seizures of duration less than one minute with the SVM correctly detecting 59.7% of seizures compared to 45.1% for GMM. Thus, the SVM system is more sensitive to short duration seizures. The GMM system however, correctly identifies more long duration seizures ($t_s > 5$ min) than the SVM.

The EEG was reviewed for each of the 147 false alarms obtained with 0.5 FD/h to determine the nature of the EEG patterns that resulted in the false detections. False detections were visually grouped into 3 classes: artifact-free background activity, artifact contaminated EEG and seizure-like activity. The occurrence of each class can be seen in Fig. 12. The background activity group was comprised of epileptiform activity and delta activity. The most prevalent artifacts causing false detections were electrode-detachment, respiration artifact and high-amplitude activity caused by movement or handling of the patient. A small proportion of the false detections

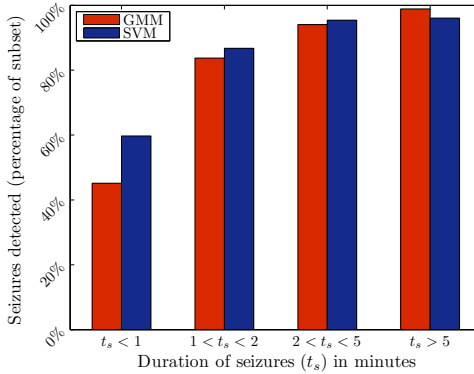


Fig. 11 Percentage seizures detected over all patients with FD/h set to 0.5. The total number of seizures in each group from lowest duration to highest is 144, 166, 218 and 177.

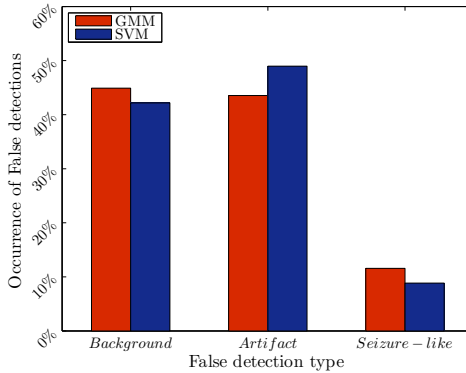


Fig. 12 Events resulting in false detections grouped according to type. Both systems are set to 0.5 FD/h resulting in 147 false detections per system.

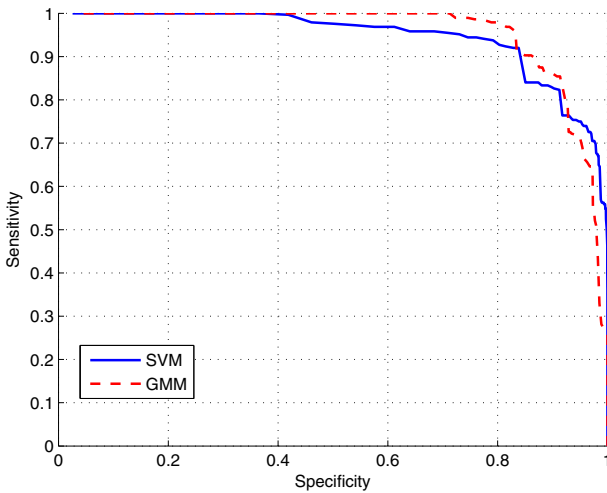
were found to correlate with seizure-like patterns. The GMM system detected 17 of these events compared to 13 for the SVM system.

4.2 Results for Individual Patients

The ROC and PR areas obtained from both systems are shown for each patient in table 2. It was shown in section 4.1 that the SVM system yields higher ROC area over all patients. However, the GMM system obtains larger ROC areas for patient 2, 12 and 16. Three patients (1, 2 and 7) show low PR area for both systems. This is in part a result of the highly unbalanced classes in the testing dataset of these patients due to a low number of short duration seizures. In these patients in particular, the SVM system achieves a larger PR area than the GMM system.

Table 2 ROC and PR area per patient

Patient	ROC area		PR area	
	SVM	GMM	SVM	GMM
1	0.915	0.88	0.412	0.328
2	0.946	0.955	0.622	0.397
3	0.969	0.963	0.927	0.925
4	0.985	0.977	0.784	0.775
5	0.915	0.915	0.713	0.715
6	0.946	0.941	0.742	0.747
7	0.98	0.951	0.317	0.09
8	0.97	0.963	0.808	0.77
9	0.955	0.958	0.974	0.974
10	0.938	0.929	0.879	0.861
11	0.993	0.985	0.979	0.97
12	0.971	0.976	0.81	0.809
13	0.974	0.971	0.903	0.888
14	0.985	0.985	0.966	0.967
15	0.978	0.976	0.942	0.938
16	0.965	0.973	0.986	0.988
17	0.988	0.983	0.972	0.969

**Fig. 13** ROC curves for patient 2

Analysis of the decisions for specific patients reveals the complementary nature of the classifiers. The ROC curves for patient 2 are presented in Fig. 13, these show that no single classifier is preferable for over conditions for this patient. In this case, the SVM system is preferable when high specificity is required, whereas the GMM is preferable when high sensitivity is desirable.

A further example of the complementary nature of the classifiers can be found in the probability of seizure plots shown for both classifiers over a section of EEG in Fig. 14. These plots are generated by taking the maximum probability of seizure over all channels for each epoch, as this channel combination is analogous to the ‘or’ operation performed after thresholding. Overall, the classifier agreement is high, with the majority of seizure events (14/16) being detected with over 20% probability of seizure by both classifiers. However, it can be seen that two seizure events show disagreement among classifiers, with only the GMM detecting event (a) and only the SVM detecting event (b).

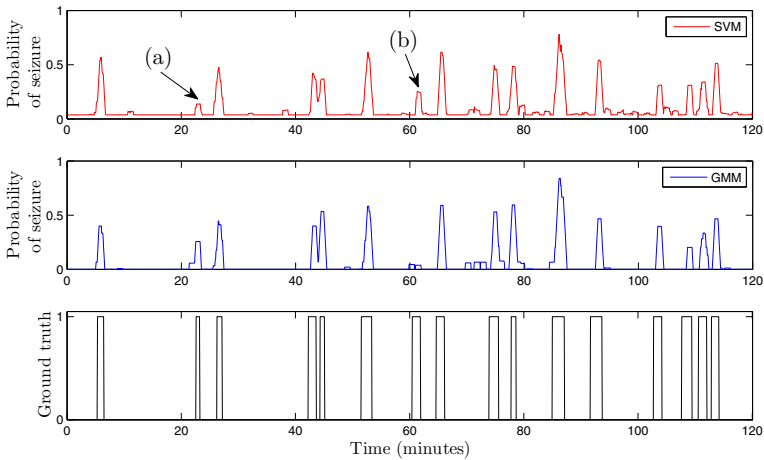


Fig. 14 SVM and GMM probabilities for a section of EEG from patient 6. Two events have been marked, these represent 2 events (a and b) where there is disagreement among classifiers.

5 Discussion

The system presented here is based on the pattern recognition framework of feature extraction followed by classification. The strength of this approach is that a large number of features can be employed to overcome the complexity of neonatal EEG. Furthermore, the non-linear classifiers are trained with background EEG data which include artifact, such that all EEG patterns are modelled within the classifier. This methodology has led to results which outperform other systems such as

Navakatikyan *et al.* (87% GDR and 2 FD/h) and Mitra *et al.* (80% GDR and 0.86 FD/h), despite the lack of a dedicated artifact removal routine. Another advantage of the system is that a soft decision may be outputted, as shown in Fig. 14, which may provide more information to the clinician than binary decisions.

The inclusion of many features, particularly power based in this study, can lead to redundancy as well as high correlations within the set. The SVM system is robust towards this feature set due to the discriminative and non-parametric nature of the classifier. For the GMM however, it was found that feature reduction techniques were required to achieve similar performance to the SVM. In [36], it was shown that either principle component analysis or LDA could be used to improve the GMM results.

The mean ROC and PR curves show that the SVM system slightly outperforms the GMM system over the epoch based metrics. The event based metric curves indicate that the SVM system detects a higher percentage of seizures than the GMM system for the same number of FD/h, although the duration of false detections is lower for the GMM. From the analysis of missed seizures in Fig. 11 it can be seen that the SVM system is more sensitive to seizures of short duration than the GMM system, thus explaining the higher GDR scores obtained by the SVM system. Both systems achieve a GDR of over 90% for all seizures of duration above 2 minutes. However, shorter duration seizures, particularly those lasting less than 1 minute, achieve lower GDR scores. It should be noted however, that the FD/h rate is set to 0.5 which can be considered a stringent operating condition and is a lower false detection rate than that reported by other systems in the literature.

The relatively low GDR scores for shorter seizures can be attributed in part to the moving average filter length. This is due to background EEG reducing the mean probability of seizure score for seizures lasting less than the length of the moving average filter. The moving average filter is 15 epochs in length, corresponding to 64 seconds, as this length was found to maximise the mean ROC area for both systems [37, 36]. It was found that misclassification of short duration seizures had a smaller adverse effect on the ROC area than misclassification of longer seizures.

Despite the reduction in sensitivity for short seizures, the postprocessing scheme used here was shown to significantly improve results in prior studies [36, 37]. The efficacy of the postprocessing routine can be explained due the assumptions present in the classifiers. The vast majority of classifiers are based on the assumption that the data is independent and identically distributed (i.i.d.). However, within a short time frame the datapoints are not independent as the EEG is a time series. Thus, it is possible to incorporate contextual information using neighbouring epochs (in time) and prior knowledge of the duration of seizure events. This type of post-processing can be seen in the works of Gotman *et al.* [27] who use a 30s gap closing procedure, Mitra *et al.* [14] who use a set of rules for growing candidate seizures and Navakatikyan *et al.* [9] who use a smoothing and gap closing procedure.

For a seizure detector to be viable for implementation in the neonatal intensive care unit, the number of false detections should be very low. From the analysis of false detections, it can be seen that false alarms are mostly due to both background activity and artifact contamination of the EEG record. The false alarms caused by

background activity were predominantly due to short runs of 'epileptiform' (periodic sharp activity) and delta activity. Epileptiform activity is a pattern resembling a seizure but which is not prolonged (less than 10 seconds) and was often found to be localized to one EEG channel. In some patients, particularly those in status epilepticus, the background EEG can alternate between these epileptiform discharges and suppressed activity for a period of time. Delta activity was found to trigger false alarms when the background patterns became more rhythmic.

Electrode detachment was found to be the most predominant cause of false detections among artifacts for both the SVM and GMM system. This artifact is characterised by a large 50Hz component as the electrode becomes contaminated with electrical noise from the environment. The 50Hz component is removed via lowpass filtering, however, other environmental signals of lower frequency are preserved in the recording and cause false alarms. Respiration artifact is repetitive in nature and thus is a cause of false alarms, along with movement artifact which causes high amplitude patterns in the EEG.

While the results from both systems are similar in terms of ROC and PR area, it is important to note that the final decisions from each system are diverse. For instance, with both systems set to 0.5 FD/h approximately half (75/147) of the false detection events do not overlap between the SVM and GMM systems outputs. Moreover, the GMM shows higher detection rates than the SVM system for seizures lasting over 5 minutes, an important subset of the seizure group as the extended length of the seizures may have implications for the seizure burden of the neonate. Furthermore, the results from individual patients showed that while the SVM system yields better ROC area in most (11/17) patients, the GMM system did perform better in some patients – in particular, patients 2, 12 and 16. It is also shown that for the same patient, the systems can produce intersecting ROC curves where each system is preferable under different operating criteria.

It was found that approximately 10% of false detections occurred from seizure-like activity, that is events for which there was uncertainty over the ground truth. In adult seizures, the inter-observer agreement was quantified by Wilson *et al.* [38], who found that the any-overlap sensitivity, i.e. GDR, was 92% and the number of false detections between observers was 0.117 per hour. To date, there have not been any studies on inter-observer agreement in neonatal seizure detection. However taking the figures obtained by Wilson *et al.*, an approximate bound on the performance of any seizure detector algorithm can be established. A number of studies report GDR scores approaching 92%, such as Navakatikyan *et al.* (87%), Deburchgraeve *et al.* (85%) and Mitra *et al.* (80%). However, the false detection rates reported by these studies – ranging from 0.66 FD/h for Deburchgraeve *et al.* to 2 FD/h for Navakatikyan *et al.* – are several times larger than the 0.117 FD/h observed by Wilson. Thus, lowering the number of false detections remains a challenge in the field.

6 Conclusion

A brief review of current trends in automated seizure detection is shown, with focus on classifier based techniques. To this end, two neonatal seizure detectors based on SVM and GMM classifiers were presented and compared. Both systems are shown to outperform previously proposed neonatal seizure detectors, indicating that classifier based methods are a viable solution for neonatal seizure detection. The SVM system produced the highest mean ROC area and mean PR area, and is therefore the better choice for classification. However, the GMM system was found to produce complementary and competitive decisions, thus warranting for research into fusion of classifiers.

Acknowledgements. This work is supported by Science Foundation Ireland (SFI/05/PICA/I836) and the Wellcome Trust (085249/Z/08/Z).

References

1. Clancy, R.: *Pediatrics* 117, 23 (2006)
2. Evans, D., Levene, M.: *Arch. Dis. Child. Fetal Neonatal Ed.* 78, 70 (1998)
3. Saliba, R.M., Annegers, J.F., Waller, D.K., Tyson, J.E., Mizrahi, E.: *American Journal of Epidemiology* 154(1), 14 (2001)
4. Bye, A.M.E., Flanagan, D.: *Epilepsia* 36(10), 1009 (1995)
5. Rennie, J.M., Chorley, G., Boylan, G.B., Pressler, R., Nguyen, Y., Hooper, R.: *Arch. Dis. Child Fetal Neonatal Ed* 89, 37 (2004)
6. De Weerd, A.W.: *Atlas of EEG in the first months of life.* Elsevier Science Ltd., Amsterdam (1995)
7. Shellhaas, R., Clancy, R.: *Clin. Neurophysiol.* 118(10), 2156 (2007)
8. Kitayama, M., Otsubo, H., Parvez, S., Lodha, A., Ying, E., Parvez, B., Ishii, R., Mizuno-Matsumoto, Y., Zoroofi, R.A., Snead, O.C.: *Pediatric Neurology* 29(4), 326 (2003)
9. Navakatikyan, M.A., Colditz, P.B., Bruke, C.J., Inder, T.E., Richmond, J., Williams, C.E.: *Clin. Neurophysiol.* 117(6), 1190 (2006)
10. Deburchgraeve, W., Cherian, P., Vos, M.D., Swarte, R., Blok, J., Visser, G., Govaert, P., Huffel, S.V.: *Clin. Neurophysiol.* 119(11), 2447 (2008)
11. Aarabi, A., Wallois, F., Grebe, R.: *Clin. Neurophysiol.* 117(2), 328 (2006)
12. Aarabi, A., Grebe, R., Wallois, F.: *Clin. Neurophysiol.* 118(12), 2781 (2007)
13. Greene, B.R., Marnane, W.P., Lightbody, G., Reilly, R.B., Boylan, G.B.: *Physiol. Meas.* 29, 1157 (2008)
14. Mitra, J., Glover, J.R., Ktonas, P.Y., Kumar, A.T., Mukherjee, A., Karayiannis, N.B., Frost, J.D., Hrachovy, R.A., Mizrahi, E.M.: *J. Clin. Neurophysiol.* 26(4), 218 (2009)
15. Schölkopf, B., Smola, A.: *Learning with Kernels.* MIT Press, Cambridge (2002)
16. Shoeb, A., Edwards, H., Connolly, J., Bourgeois, B., Treves, S.T., Gutttag, J.: *Epilepsy and Behaviour* 5, 483 (2004)
17. AcIr, N., Güzelis, C.: *Computers in Biology and Medicine* 34(7), 561 (2004)
18. Gardner, A.B., Krieger, A.M., Vachtsevanos, G., Litt, B.: *Journal of Machine Learning Research* 7, 1025 (2006)
19. Runarsson, T., Sigurdsson, S.: *Computational Intelligence for Modelling. Control and Automation* 2, 673 (2005)
20. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: *Digital Signal Processing* 10(1-3), 19 (2000)

21. Zhu, X., Wu, J., Cheng, Y., Wang, Y.: Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006), pp. 1171–1174 (2006)
22. Marcel, S., Millan, J.: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 29(4), 743 (2007)
23. Meng, L., Frei, M., Osorio, I., Strang, G., Nguyen, T.: *Med. Eng. Phys.* 26(5), 379 (2004)
24. Vapnik, V.: *Estimation of Dependences Based on Empirical Data*. Springer, New York (1982)
25. Greene, B.R., Faul, S., Marnane, W.P., Lightbody, G., Korotchikova, I., Boylan, G.B.: *Clin. Neurophysiol.* 119(6), 1248 (2008)
26. Shoeb, A., Bourgeois, B., Treves, S.T., Schachter, S.C., Gutttag, J.: In: *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, pp. 4110–4114 (2007)
27. Gotman, J., Flanagan, D., Zhang, J., Rosenblatt, B.: *Electroenceph. clin. Neurophysiol.* 103, 356 (1997)
28. Hjorth, B.: *Electroencephalogr. Clin. Neurophysiol.* 29(3), 306 (1970)
29. D'Alessandro, M., Esteller, R., Vachtsevanos, G., Hinson, A., Echauz, J., Litt, B.: *IEEE Trans. Biomed. Eng.* 50, 603 (2003)
30. Esteller, R., Echauz, J., Tchong, T., Litt, B., Pless, B.: Proceedings of the 23rd Annual EMBS International Conference, pp. 1707–1710 (2001)
31. Faul, S., Boylan, G.B., Connolly, S., Marnane, W.P., Lightbody, G.: Proceedings of the IEEE International Symposium on Intelligent Signal Processing, pp. 381–386 (2005)
32. Platt, J.: *Advances in large margin classifiers*, 61–74 (1999)
33. Dempster, A.P., Laird, N.M., Rubin, D.B.: *Journal of the Royal Statistical Society. Series B (Methodological)* 39(1), 1 (1977)
34. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. Wiley-Interscience, Hoboken (2001)
35. Duchene, J., Leclercq, S.: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 10(6), 978 (1988)
36. Thomas, E., Temko, A., Lightbody, G., Marnane, W., Boylan, G.: *IEEE MLSP* (2009)
37. Temko, A., Thomas, E., Boylan, G., Marnane, W., Lightbody, G.: *IEEE EMBC* (2009)
38. Wilson, S.B., Scheuer, M., Plummer, C., Young, B., Pacia, S.: *Clin. Neurophysiol.* 2156(114) (2003)

Design of Fuzzy Relation-Based Image Sharpeners

Fabrizio Russo

Abstract. Fuzzy relations among pixel luminances are simple and effective tools for the processing of digital images. This chapter shows how fuzzy relations can be adopted in the design of a complete image enhancement systems and successfully address conflicting tasks such as detail sharpening and noise cancellation. For this purpose, the different behaviors of fuzzy relation-based high-pass filters and noise smoothers are explained along with the effects of different parameter settings. Results of computer simulations show that fuzzy relation-based processing is an effective resource for the sharpening of noisy images and is easy to use.

Keywords: fuzzy models, fuzzy relations, image sharpening, noise cancellation, detail preservation, image quality assessment.

1 Introduction

Digital methods for image sharpening [1] are adopted in a growing number of research and application areas such as robotics, medical systems, remote sensing, video surveillance and biometrics, where digital images have become a primary source of information. Indeed, contrast enhancement aims at highlighting important features embedded in the image data, so it can significantly increase the accuracy of subsequent processing tasks such as parameter estimation, object recognition and scene interpretation. Since contrast enhancement typically improves the visual quality of a picture, it is widely used in consumer electronics dealing with digital cameras and camcorders.

It is known, however, that a very critical issue in the enhancement of digital images is the noise increase generated by the sharpening process. The linear unsharp masking (UM) technique is a classical example, where the effect of the noise is very annoying [2]. In this method, a fraction of the high-pass filtered

Fabrizio Russo
D.E.E.I. – University of Trieste, Trieste, Italy

image is added to the original data producing edge enhancement and noise amplification as well. In order to address this issue, more effective approaches have resorted to nonlinear operators that can achieve a better compromise between detail sharpening and noise amplification [3-7]. In this framework, significant results were obtained by adopting weighted medians (WM) and permutation weighted medians (PWMs) as a replacement for high-pass linear filters in the UM scheme, because such nonlinear operators can successfully limit the noise amplification produced by the sharpening process [8-10]. Polynomial UM methods were also investigated. Very interesting examples of this class of nonlinear enhancement techniques are the Teager-based operator [11-12] and the cubic UM method [13-14]. Rational UM operators have shown to be effective for contrast enhancement of digital images [15]. These nonlinear techniques can avoid noise increase and excessive overshoot on object contours. Nonlinear methods based on fuzzy models have also been proposed in the literature. Indeed, fuzzy systems are well suited to model the uncertainty that occurs when conflicting tasks are performed, for example, detail sharpening and noise cancellation [16-19]. In this respect, it should be observed that even if rule-based systems are very effective tools, they are not the only way to process digital pictures. Image processing algorithms based on fuzzy relations can better conjugate simplicity and effectiveness [20].

The aim of this chapter is to describe how fuzzy relations can be adopted to perform sharpening of noisy digital images. For this purpose, after a brief review of the classical linear UM approach, we shall present the basic scheme of a nonlinear UM operator adopting a fuzzy relation-based high-pass filter. The behavior of this operator will be further analyzed by taking into account how edge enhancement and sensitivity to noise depend upon the appropriate choice of fuzzy relation shape. The properties of fuzzy relation-based smoothers for the reduction of Gaussian noise and the removal of outliers will be discussed too. Finally, the design of a complete contrast enhancement system incorporating the mentioned algorithms will be presented. Results of computer simulations will be reported to show the effectiveness of the proposed method.

2 Linear Unsharp Masking: A Brief Review

As aforementioned, the general UM scheme resorts to a high-pass filter in order to highlight edges and fine details of a picture. Formally, let us suppose we deal with digitized images having L gray levels. Let $x(i,j)$ be the pixel luminance at location $[i,j]$ in the input image and let $h(i,j)$ be the pixel luminance at the same location in the high-pass filtered picture. Thus, an image sharpener based on the UM approach can be defined as follows:

$$y(i,j) = x(i,j) + \lambda h(i,j) \quad (1)$$

where $y(i,j)$ denotes the pixel luminance at location $[i,j]$ in the sharpened image and λ is a weighting factor. The classical UM method resorts to a linear high-pass filter, such as the following operator:

$$h(i, j) = \sum_{x(m,n) \in W_0} x(i, j) - x(m, n) \quad (2)$$

where W_0 is the set of neighboring pixels: $\{x(i-1, j), x(i+1, j), x(i, j-1), x(i, j+1)\}$.

Typically, high-pass filtering increases the noise as shown in Fig.1. In this example, we considered the original "Lena" picture as input image (Fig.1a). The enhanced and the high-pass filtered data are depicted in Fig.1b and Fig.1c, respectively. The noise amplification yielded by the linear sharpener is clearly perceivable, especially if we look at the uniform regions of the picture. In order to focus on the noise sensitivity of the method, let us consider a second example where we corrupted the "Lena" image by adding zero-mean Gaussian noise with standard deviation $\sigma=5$ (Fig.2a). It can be seen that the noise increase is very annoying (Fig.2b). Indeed, linear filters cannot distinguish between detail enhancement and noise amplification (Fig.2c), whereas nonlinear operator can. We shall describe in the next section how fuzzy relations among pixel luminances can be adopted to design pseudo high-pass filters that can offer a very easy control of the sharpening action.

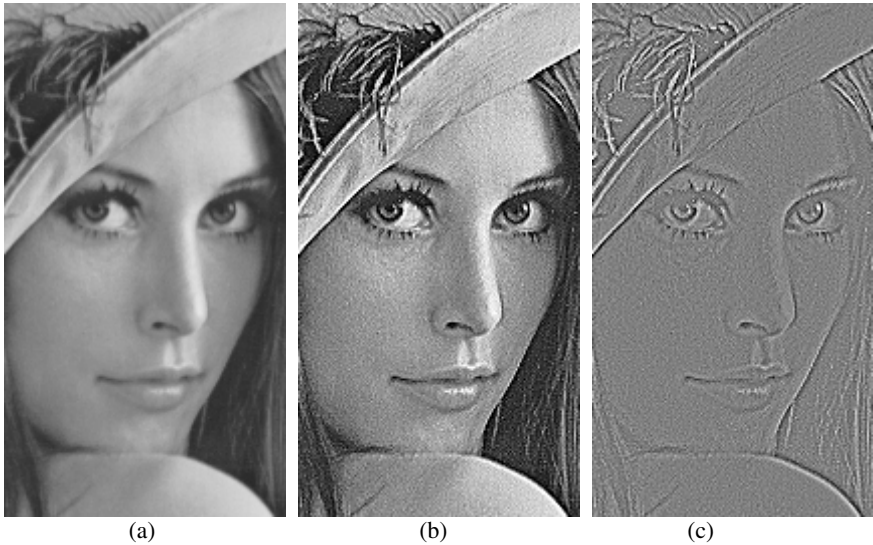


Fig. 1 Portion of the original 512×512 "Lena" picture (a), result of linear UM (b), high-pass component (c)

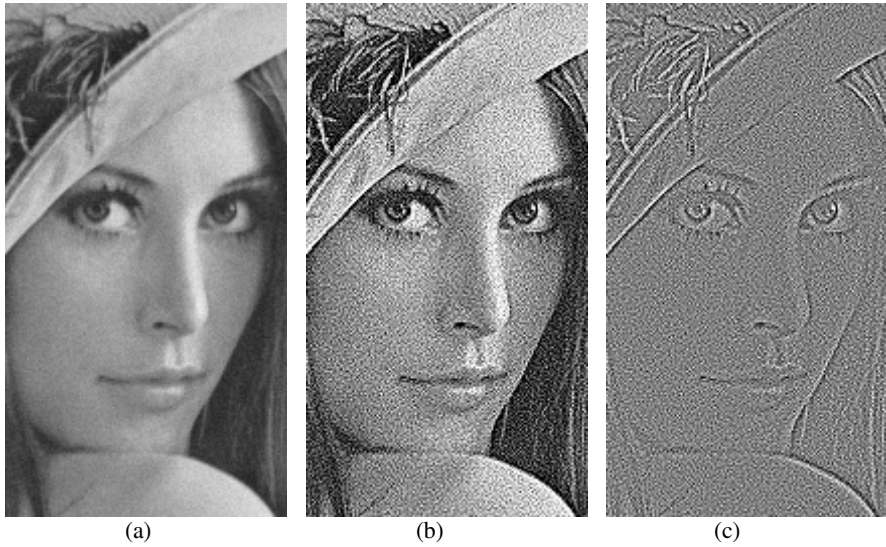


Fig. 2 “Lena” picture corrupted by Gaussian noise with standard deviation $\sigma=5$ (a), result of linear UM (b), high-pass component (c)

3 Nonlinear Unsharp Masking Based on Fuzzy Relations

Since sharpening aims at highlighting the luminance differences among pixels, fuzzy relations like “ $x(m,n)$ is different from $x(i,j)$ ” are the appropriate choice. Thus, according to the preceding considerations, we can define a fuzzy relation-based UM operator as follows:

$$y(i, j) = x(i, j) + \lambda h_1(i, j) \quad (3)$$

$$h_1(i, j) = \frac{1}{N} \sum_{x(m,n) \in W} [x(i, j) - x(m, n)] \mu_{DIF}[x(i, j), x(m, n)] \quad (4)$$

where $h_1(i, j)$ is the output of the pseudo high-pass filter, W denotes a set of N neighboring pixels around $x(i, j)$ and $\mu_{DIF}(u, v)$ is the parameterized membership function that describes the fuzzy relation “ u is different from v ”:

$$\mu_{DIF}(u, v) = \begin{cases} 0 & |u - v| < a \\ \frac{b(|u - v| - a)}{|u - v|(b - a)} & a \leq |u - v| < b \\ 1 & b \leq |u - v| < c \\ \frac{c}{|u - v|} & |u - v| \geq c \end{cases} \quad (5)$$

where a , b and c are parameters ($0 < a < b < c$). An example of graphical representation of the function $\mu_{DIF}(u,v)$ is reported in Fig.3.

Clearly, the membership function shape plays a key role in providing the correct behavior of the sharpener. In particular, the sensitivity to noise is controlled by the parameter a whereas the responses to small-medium and strong edges depend upon the parameters b and c , respectively. Indeed, the membership function

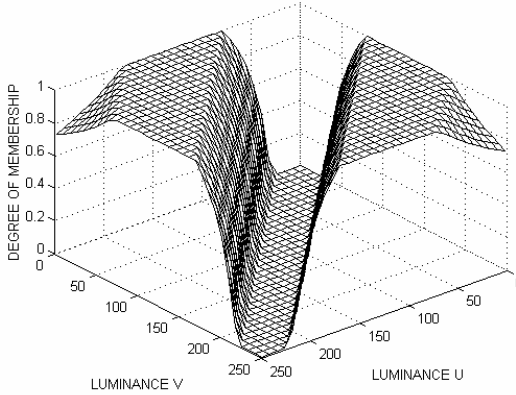


Fig. 3. Example of graphical representation of the membership function $\mu_{DIF}(u,v)$

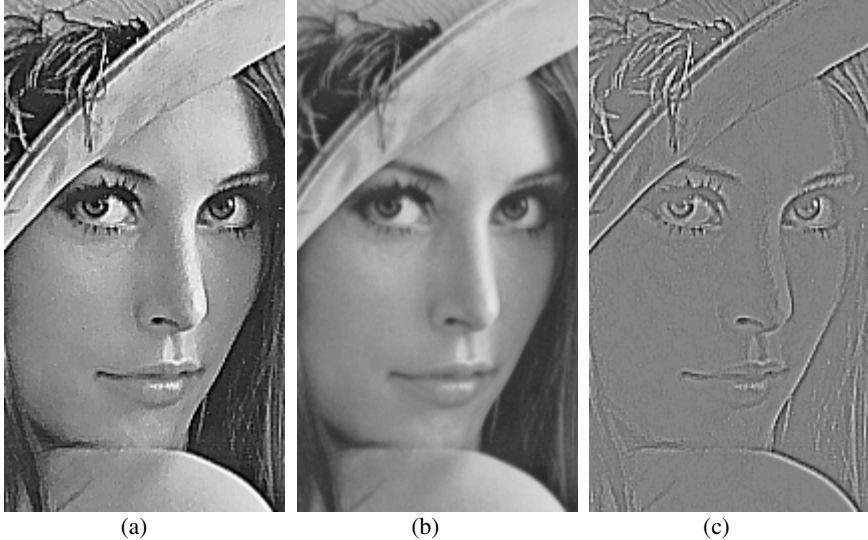


Fig. 4 (a) Portion of the original 512×512 “Lena” picture, (b) result of fuzzy relation-based UM ($a=4$, $b=40$, $c=80$, $\lambda=5$), (c) high-pass component

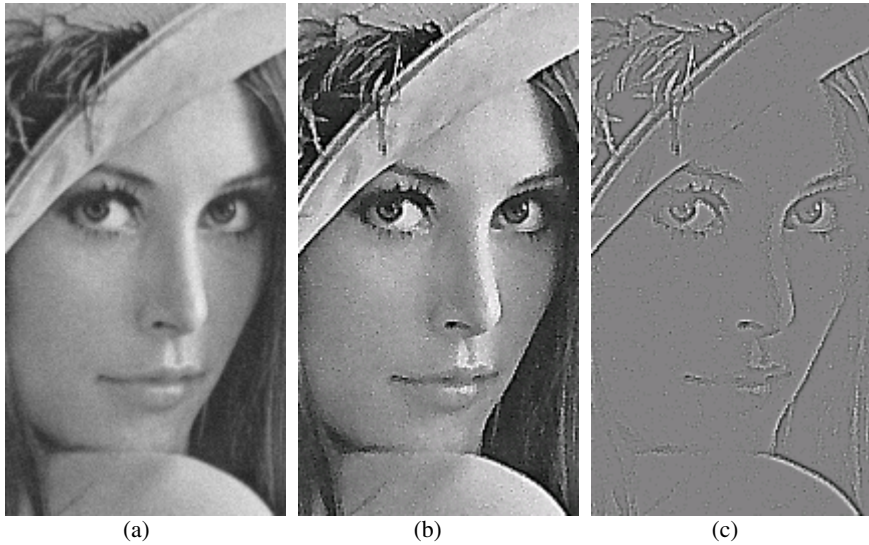


Fig. 5 (a) “Lena” picture corrupted by Gaussian noise with standard deviation $\sigma=5$, (b) result of fuzzy relation-based UM ($a=15$, $b=40$, $c=80$, $\lambda=5$), (c) high-pass component

shape is designed to limit the noise increase, to strongly enhance small-medium edges and to moderately enhance large edges, in order to avoid an excess of overshoots along the object contours. The overall amount of sharpening is easily controlled by the parameter λ .

Two application examples of the operator dealing with the neighborhood $W = \{x(i-1, j-1), x(i-1, j), x(i-1, j+1), x(i, j-1), x(i, j+1), x(i+1, j-1), x(i+1, j), x(i+1, j+1)\}$ are reported in Fig.4 and 5 for visual inspection. They can be compared to the results yielded by the linear UM method for original (Fig 1) and noisy (Fig 2) input data. The noise increase given by the nonlinear technique is significantly smaller in both cases.

4 Effects of Parameter Settings

We shall briefly describe in this section the effects of different parameter settings. According to (3)–(5), the overall behavior of the fuzzy sharpener is controlled by four parameters λ , a , b and c .

The role played by λ is very simple: it defines the weight of the high-pass component in eq.(3), i.e. the amount of the high-pass filtered data added to the input image. Thus the sharpening action becomes stronger as the value of λ increases. Too large values should be avoided because they typically produce excessive overshoots on object borders and amplification of the residual noise. The effects of different parameter choices are shown in Fig.6.

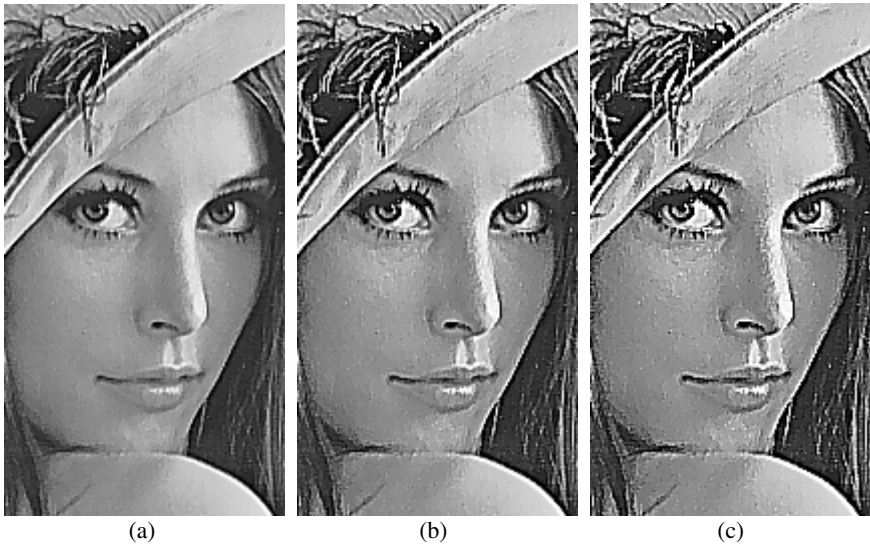


Fig. 6 Sharpening of the original “Lena” image and effects of different choices of λ ($a=4$, $b=50$, $c=100$): (a) $\lambda=3$, (b) $\lambda=5$, (c) $\lambda=7$

The sensitivity to noise is mainly controlled by the parameter a . In this approach, small luminance differences ($|x(i,j)-x(m,n)| < a$) are considered as noise, thus no sharpening is performed ($\mu_{DIF}=0$). The value of this parameter should be carefully chosen.

Too small values increase the noise whereas too large values limit noise amplification and enhancement of fine details as well. Fig.7 shows examples of wrong and correct settings for this parameter when the input image is the “Lena” picture corrupted by Gaussian noise with $\sigma=5$. The choice $a=\sigma$ (Fig.7a) is quite unsatisfactory because the resulting image is very noisy (Fig.7b). Indeed, this value is too small and the sharpening is activated for all pixels having noise amplitude larger than σ . The choice $a=3\sigma$ (Fig.7c) is much more appropriate. It can be seen (Fig.7d) that the image is significantly less noisy. Increasing the value of this parameter would further limit the noise amplification. In this case, however, more details would remain unprocessed, i.e. without any sharpening.

The role played by parameters b and c is much less critical. Indeed, when the luminance differences are not small ($|x(i,j)-x(m,n)| \geq a$) the sharpening effect is activated with variable strength, depending on the slope of the membership function μ_{DIF} . In particular, when the luminance differences are small-medium, i.e. when $a < |x(i,j)-x(m,n)| \leq b$, the output yields strong sharpening in order to highlight image details. Since an excess of overshoots along the object borders would be annoying, the sharpening is limited when the luminance differences are large ($b < |x(i,j)-x(m,n)| \leq c$) and further reduced when the luminance differences are very large ($|x(i,j)-x(m,n)| \geq c$).

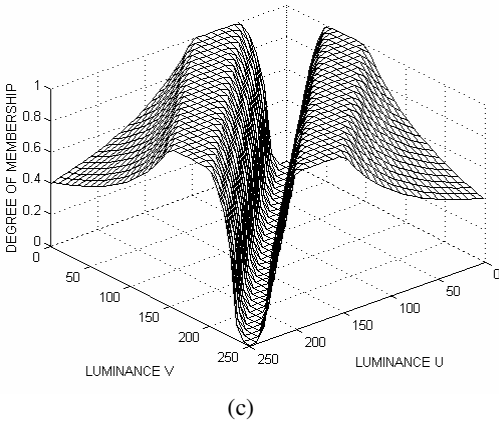
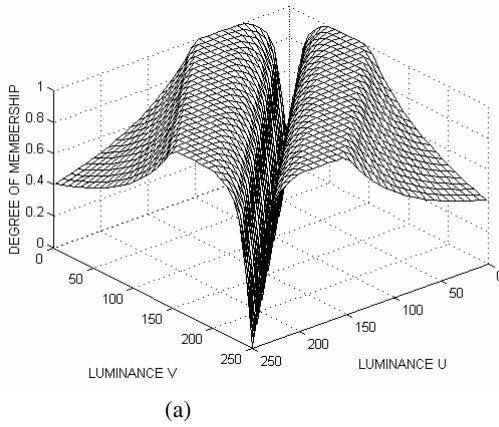


Fig. 7 Effects of different choices of the parameter a when the input image is the “Lena” picture corrupted by Gaussian noise with $\sigma=5$: (a) shape of μ_{DIF} ($a=5$, $b=50$, $c=100$), (b) result ($\lambda=6$), (c) shape of μ_{DIF} ($a=15$, $b=50$, $c=100$), (d) result ($\lambda=6$).

5 Noise Prefiltering Using Fuzzy Relations

As described in the previous section, the fuzzy relation-based operator can effectively limit the noise amplification during sharpening. However, it cannot reduce the noise. In order to perform this task, a prefiltering action is necessary. We shall

focus on noise having Gaussian-like distribution because it is very often encountered in real images due to noisy sensors and/or high sensitivity settings during image acquisition. Furthermore, we shall assume that the amount of noise corruption is limited ($\sigma < 10$). The proposed prefiltering is based on classification of noisy pixels into two different (fuzzy) classes [21]:

- 1) pixels corrupted by noise whose amplitude is similar to that of the neighbors (class A pixels);
- 2) pixels corrupted by noise whose amplitude is much larger than that of the neighbors (class B pixels).

A – Fuzzy relation-based prefiltering of type A pixels: basic design

Since this kind of prefiltering deals with class A pixels, fuzzy relations like “ $x(m,n)$ is similar to $x(i,j)$ ” represent the appropriate choice. Thus, a simple fuzzy relation-based noise smoother can be defined as follows:

$$y(i, j) = x(i, j) - g(i, j) \quad (6)$$

$$g(i, j) = \frac{1}{8} \sum_{x(m,n) \in W} [x(i, j) - x(m, n)] \mu_{SIM} [x(i, j), x(m, n), p, q] \quad (7)$$

where $g(i, j)$ is an estimate of the noise amplitude at location $[i, j]$ and $\mu_{SIM}(u, v, p, q)$ is the parameterized membership function that describes the fuzzy relation “ u is similar to v ”:

$$\mu_{SIM}(u, v, p, q) = \begin{cases} 1 & |u - v| < p \\ \frac{pq - |u - v|}{(q - 1)|u - v|} & p \leq |u - v| < pq \\ 0 & |u - v| \geq pq \end{cases} \quad (8)$$

where p and q are parameters ($0 < p < L, q > 1$). Fig.8 shows the influence of different parameter settings on the membership function shape. The operation defined by (6-7) is very simple: the processing takes into great account small luminance differences (possibly caused by type A noise) and excludes large luminance differences representing object borders in order to avoid image blurring. When all the absolute differences between the central pixel and its neighbors are smaller than p , only noise is assumed to be present. Thus a strong smoothing is performed and the result is the arithmetic mean of the pixel luminances in the neighborhood. In this model, differences larger than pq denote edges and then their contribution is zero. Indeed, when $|x(i, j) - x(m, n)| < p$ we have $\mu_{SIM} = 1$ (type A pixels). Conversely, when $|x(i, j) - x(m, n)| \geq pq$ we have $\mu_{SIM} = 0$ (object border). The shape of the membership function μ_{SIM} is designed to perform a gradual transition between these opposite

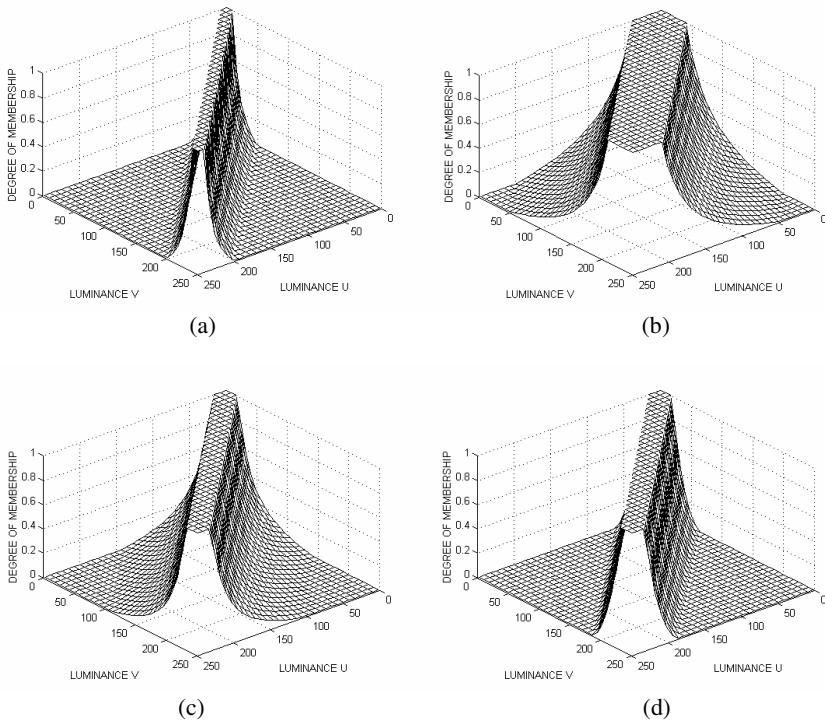


Fig. 8 Examples of graphical representation of the membership function $\mu_{SIM}(u, v, p, q)$: (a) $p=10$, $q=5$; (b) $p=40$, $q=5$; (c) $p=20$, $q=3$; (d) $p=20$, $q=7$

effects when the luminance differences are medium $p \leq |x(i, j) - x(m, n)| < pq$. The parameters p and q offer great flexibility in defining the filtering action. By suitably choosing these parameters, we can decide how much a luminance difference should be considered useful information or unwanted noise. According to these observations, large values of p are typically required in the presence of large noise variances. These values increase the smoothing effect at the price of a lower detail preservation. The role of the parameter q is less critical: satisfactory values can be found in the range $3 \leq q < 8$.

B – Fuzzy relation-based prefiltering of type A pixels: advanced design

We shall briefly describe here a more advanced scheme that can better adapt the filtering behavior to the local characteristics of the image. For this purpose, we shall consider the spatial and amplitude relationships between the central pixel in the moving window and its neighbors [22]. Let the neighboring pixels be grouped into two different subsets:

$$W_1 = \{x(i-1, j), x(i, j-1), x(i, j+1), x(i+1, j)\},$$

$$W_2 = \{x(i-1, j-1), x(i-1, j+1), x(i+1, j-1), x(i+1, j+1)\}.$$

Thus, let us define the filter’s output $y(i, j)$ by resorting to *two* fuzzy relations dealing with the inner and outer subsets, respectively:

$$y(i, j) = x(i, j) - g(i, j) \tag{9}$$

$$g(i, j) = \frac{1}{4} \sum_{x(m,n) \in W_1} [x(i, j) - x(m, n)] \mu_{SIM} [x(i, j), x(m, n), p_1, q_1] + \frac{1}{4} \sum_{x(m,n) \in W_2} [x(i, j) - x(m, n)] \mu_{SIM} [x(i, j), x(m, n), p_2, q_2] \tag{10}$$

The filtering defined by (9-10) can be adapted with different strengths to two subsets W_1 and W_2 in order to exploit the spatial information. The pixels in the set W_1 are closer (and then more correlated) to $x(i, j)$ than the pixels in W_2 . Thus, a more accurate filtering can be performed by applying stronger smoothing in W_1 and weaker smoothing (i.e. stronger detail preservation) in W_2 . This goal is achieved by setting $p_2 < p_1$ (Fig.9).

Satisfactory results can be achieved when p_1 is the only parameter that depends upon the noise variance and $q_1, q_2, p_2/p_1$ are constant values.

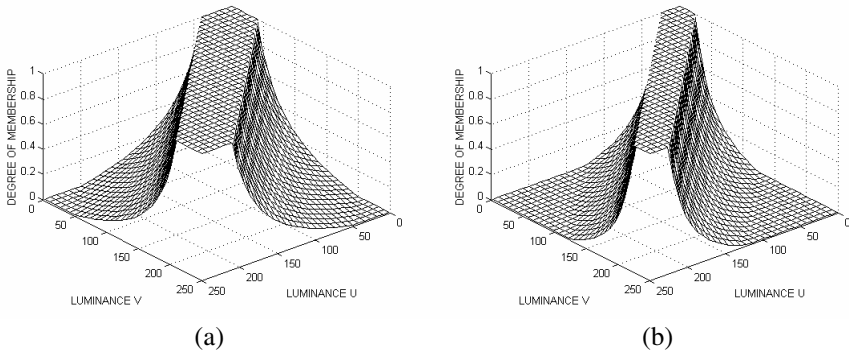


Fig. 9 Examples of membership function choices for different pixel subsets: (a) $\mu_{SIM}(u, v, p_1, q_1)$ for subset $W_1 = \{x(i-1, j), x(i, j-1), x(i, j+1), x(i+1, j)\}$; (b) $\mu_{SIM}(u, v, p_2, q_2)$ for subset $W_2 = \{x(i-1, j-1), x(i-1, j+1), x(i+1, j-1), x(i+1, j+1)\}$.

C – Fuzzy relation-based prefiltering of type B pixels

We shall focus here on type B pixels that typically represent outliers, i.e., large amplitude noisy pixels present in the data as an effect of the “tail” of the Gaussian distribution.

Type B noise prefiltering addresses the luminance differences between the central pixel and its neighbors in a different way: if all these differences are very large, the pixel is (possibly) an outlier to be cancelled. Clearly, fuzzy relations like “ $x(i,j)$ is larger than $x(m,n)$ ” would be very helpful for this purpose. Thus, a simple fuzzy relation-based filter for the removal of type B noise can be defined as follows:

$$y(i, j) = x(i, j) - g_b(i, j) \tag{11}$$

$$g_b(i, j) = (L - 1) \frac{\text{MIN}_{x(m,n) \in W_0} \{ \mu_{LA}(x(i, j), x(m, n)) \} - \text{MIN}_{x(m,n) \in W_0} \{ \mu_{LA}(x(m, n), x(i, j)) \}}{L - 1} \tag{12}$$

where $\mu_{LA}(u,v)$ is the membership function that describes the fuzzy relation “ u is larger than v ”:

$$\mu_{LA}(u,v) = \begin{cases} \frac{u - v}{L - 1} & 0 < u - v \leq L - 1 \\ 0 & u - v \leq 0 \end{cases} \tag{13}$$

The shape of $\mu_{LA}(u,v)$ is designed to yield a perfect correction of the noise in the ideal case (Fig.10). Indeed, let W_0 be a perfectly uniform neighborhood formed by pixels having the same luminance x_c and let $x(i,j) \gg x_c$ be a positive outlier. According to (12) we have $g_b(i,j) = x(i,j) - x_c$ and thus $y(i,j) = x_c$. The outlier has been cancelled.

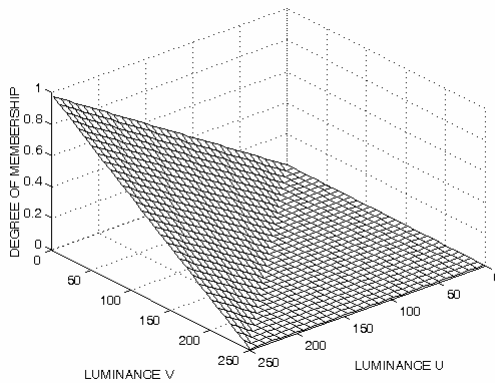


Fig. 10 Graphical representation of the membership function $\mu_{LA}(u,v)$

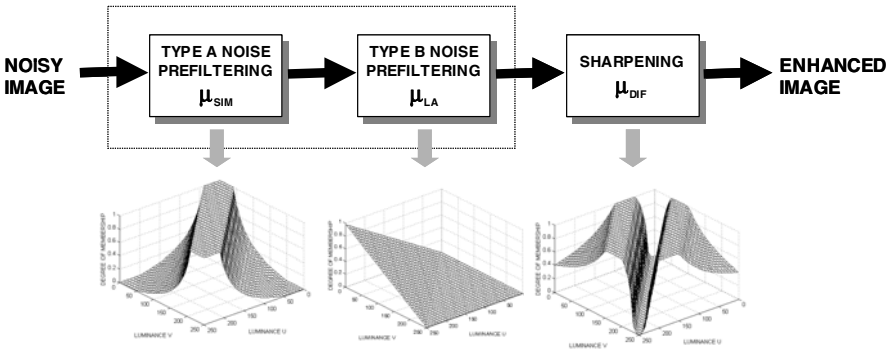


Fig. 11 Block diagram of the complete image enhancement system adopting different fuzzy relations for smoothing and sharpening

6 A Complete Fuzzy Relation-Based Image Enhancement System

The block diagram of a complete enhancement system adopting fuzzy relations for noise smoothing and image sharpening is shown in Fig.11. The system is composed of three cascaded modules.

The first processing block performs prefiltering of type A noisy pixels, according to (9-10). The second module removes type B noisy pixels according to (11-12). Finally, the third block performs sharpening of the (possibly) noise-free resulting picture (see eq.(3-4)). Fig.11 highlights the different role played by membership functions μ_{SIM} , μ_{LA} , μ_{DIF} in the overall processing. The choice of a satisfactory shape for μ_{SIM} is not a difficult task. As mentioned in Section 5, good results can be obtained if we choose p_1 according to the noise variance and we adopt constant values for q_1 , q_2 and p_2/p_1 . Table I shows an example of parameter assignment. In this example, we considered the “Lena” picture corrupted by Gaussian noise with standard deviation ranging from 4 to 10. We chose $q_1=q_2=6$ and $p_2/p_1=0.5$ in all cases. The table reports (second column) the values of p_1 that yield the minimum mean squared error (MSE) between the original image and the processed data. The prefiltering performance with respect to noise cancellation and detail preservation can be assessed, without the need for visual inspection

Table.1 Parameter assignment and MSE values (“Lena” picture corrupted by Gaussian noise with standard deviation σ ranging from 4 to 10)

σ	p_1	MSE	MSE _{RN}	MSE _{CD}
4	5	8.80	3.785	5.016
6	9	14.35	6.166	8.184
8	12	20.65	8.585	12.060
10	16	27.72	11.270	16.447

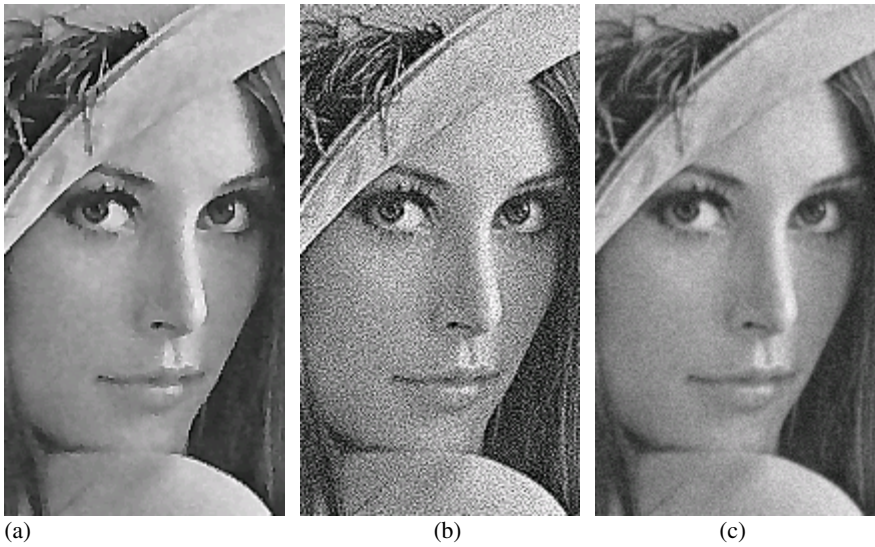


Fig. 12 (a) Noisy input image ($\sigma=8$), (b) result given by linear UM, (c) result given by the complete fuzzy relation-based image enhancement system ($p_1=12$, $a=14$, $b=50$, $c=60$, $\lambda=3$).

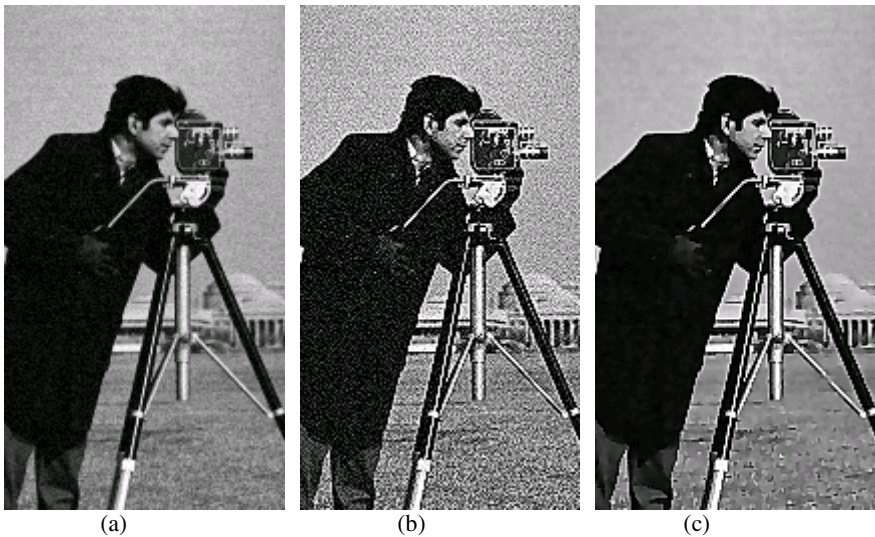


Fig. 13 (a) Noisy input image ($\sigma=8$), (b) result given by linear UM, (c) result given by the complete fuzzy relation-based image enhancement system ($p_1=12$, $a=14$, $b=50$, $c=60$, $\lambda=3$).

(and with more accuracy), by resorting to a recently introduced method for objective evaluation of such features [23]. In this approach, the correctness of the filtering is analyzed and this information is exploited to decompose the MSE into two components that respectively measure the *residual noise* due to insufficient filtering (MSE_{RN}) and the *collateral distortion* due to excessive or wrong filtering (MSE_{CD}). The values of these MSE components are also reported in Tab.I to characterize the behavior of the adopted fuzzy relation-based prefiltering.

A first application example of the overall enhancement system is shown in Fig.12. In this example we considered the “Lena” picture corrupted by Gaussian noise with standard deviation $\sigma=8$ (Fig.12a). The result given by the linear UM technique is reported in Fig.12b for a comparison. The sharpened image is very noisy and the result is quite unsatisfactory. The result yielded by the proposed fuzzy relation-based enhancement system is shown in Fig.12c. We prefiltered the noisy input image according to the data in Tab.I, i.e., by choosing $p_1=12$ ($q_1=q_2=6$, $p_2/p_1=0.5$). We adopted the following settings for the sharpening parameters: $a=14$, $b=50$, $c=60$, $\lambda=3$. It can be seen that the sharpened image is significantly less noisy than the input picture and the details look very sharp.

A second application example dealing with the 256×256 “Cameraman” picture is reported in Fig.13. We corrupted the image by using the same amount of Gaussian noise as in the previous example ($\sigma=8$). We also adopted the same parameter settings for the overall enhancement systems in order to test the robustness of the approach. The effectiveness of the fuzzy enhancement can be appraised if we observe the image in Fig.13c and, especially, if we compare it to the result given by the linear UM technique (Fig13b).

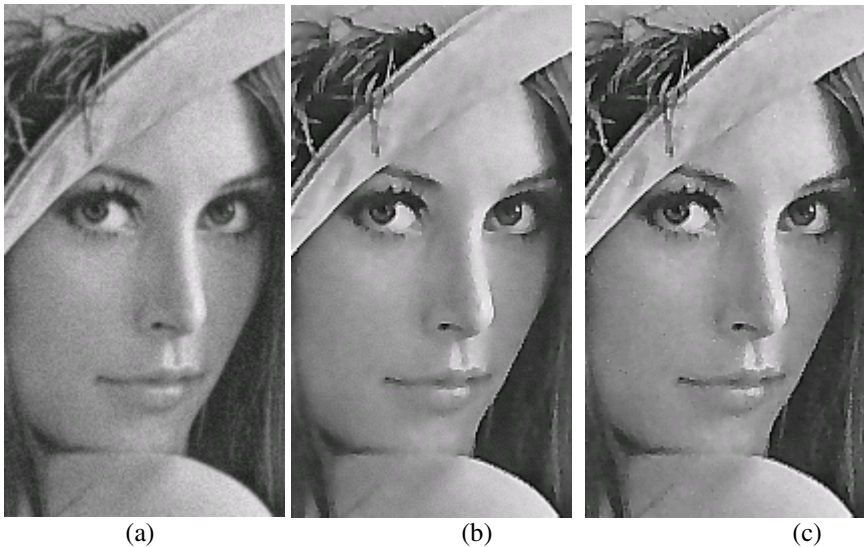


Fig. 14 (a) Noisy input image ($\sigma=6$), (b) result given by $\{ p_1=12, q_1=q_2=6, p_2/p_1=0.5, a=14, b=50, c=60, \lambda=3 \}$, (c) result given by $\{ p_1=9, q_1=q_2=6, p_2/p_1=0.5, a=11, b=50, c=60, \lambda=3 \}$

In a third example, we considered the “Lena” picture corrupted by Gaussian noise with standard deviation: $\sigma=6$ (Fig.14a) and the results given by two slightly different sets of parameter values: $\{ p_1=12, q_1=q_2=6, p_2/p_1=0.5, a=14, b=50, c=60, \lambda=3 \}$ (Fig.14b) and $\{ p_1=9, q_1=q_2=6, p_2/p_1=0.5, a=11, b=50, c=60, \lambda=3 \}$ (Fig.14c). We can see that the differences in the parameter settings are not very critical and satisfactory results are achieved in both cases (the image in Fig14c represents the correct choice).

7 Conclusion

In this chapter we have described how fuzzy relation-based operators can constitute the key components of an image enhancement system. First, we have presented the basic scheme of a nonlinear UM operator including a fuzzy relation-based high-pass filter. We have analyzed the behavior of this operator by considering how edge enhancement and sensitivity to noise depend upon the appropriate choice of the membership function shape. Thus, we have described different kinds of noise smoothers where fuzzy relations are adopted to reduce noise while preserving the image details. Finally, we have presented the design of a complete contrast enhancement system combining fuzzy relation-based smoothing and sharpening. Computer simulations have shown that the proposed method gives very satisfactory results and that the choice of parameter values is easy.

References

1. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Pearson International, London (2008)
2. Jain, A.K.: Fundamentals of Digital Image Processing. Prentice-Hall, Englewood Cliffs (1989)
3. Ramponi, G.: Polynomial and rational operators for image processing and analysis. In: Mitra, S.K., Sicuranza, G. (eds.) Nonlinear Image Processing, pp. 203–223. Academic, London (2000)
4. Arce, G.R., Paredes, J.L.: Image enhancement and analysis with weighted medians. In: Mitra, S.K., Sicuranza, G. (eds.) Nonlinear Image Processing, pp. 27–67. Academic, London (2000)
5. Matz, S.C., de Figueiredo, R.J.P.: A nonlinear technique for image contrast enhancement and sharpening. In: Proc. IEEE ISCAS, pp. 175–178 (1999)
6. De Figueiredo, R.J.P., Matz, S.C.: Exponential nonlinear Volterra filters for contrast sharpening in noisy images. In: Proc. IEEE ICASSP, pp. 2263–2266 (1996)
7. Polesel, A., Ramponi, G., Mathews, V.J.: Image enhancement via adaptive unsharp masking. IEEE Trans. Image Process. 9(3), 505–510 (2000)
8. Hardie, R.C., Barner, K.E.: Extended permutation filters and their application to edge enhancement. IEEE Trans. Image Process. 5(6), 855–867 (1996)
9. Fischer, M., Paredes, J.L., Arce, G.R.: Image sharpening using permutation weighted medians. In: Proc. X EUSIPCO, Tampere, Finland, pp. 299–302 (2000)
10. Fischer, M., Paredes, J.L., Arce, G. R.: Weighted median image sharpeners for the World Wide Web. IEEE Trans. Image Process. 11(7), 717–727 (2002)

11. Mitra, S.K., Li, H., Lin, I.-S., Yu, T.-H.: A new class of nonlinear filters for image enhancement. In: Proc. Int. Conf. Acoust., Speech Signal Process, Toronto, ON, Canada, pp. 2525–2528 (1991)
12. Thurnhofer, S.: Two-dimensional teager filters. In: Mitra, S.K., Sicuranza, G. (eds.) *Nonlinear Image Processing*, pp. 167–202. Academic, London (2000)
13. Ramponi, G., Strobel, N., Mitra, S.K., Yu, T.-H.: Nonlinear unsharp masking methods for image contrast enhancement. *J. Electron. Imaging* 5(3), 353–366 (1996)
14. Nakashizuka, M., Aokii, I.: A cascade configuration of the cubic unsharp masking for noisy image enhancement. In: Proc. Int. Symp. Intell. Signal Process. Commun. Syst., Hong Kong, pp. 161–164 (2005)
15. Ramponi, G., Polesel, A.: A rational unsharp masking technique. *J. Electron. Imaging* 7(2), 333–338 (1998)
16. Russo, F.: An image enhancement technique combining sharpening and noise reduction. *IEEE Trans. Instrum. Meas.* 51(4), 824–828 (2002)
17. Russo, F.: An Image Enhancement System Based on Noise Estimation. *IEEE Trans. Instrum. Meas.* 56(4), 1435–1442 (2007)
18. Rovid, A., Varkonyi-Koczy, A.R., Varlaki, P.: 3D Model Estimation from Multiple Images. In: Proc. FUZZ-IEEE 2004, Budapest, Hungary (2004)
19. Rovid, A., Varkonyi-Koczy, A.R., Da Graca Ruano, M., Varlaki, P., Michelberger, P.: Soft Computing Based Car Body Deformation and EES Determination for Car Crash Analysis Systems. In: Proc. IMTC 2004, Como, Italy (2004)
20. Russo, F.: Fuzzy Models for Low-Level Computer Vision: A Comprehensive Approach. In: Proc. 2007 IEEE International Symposium on Intelligent Signal Processing, WISP 2007, Alcalà de Henares, Madrid, Spain (2007)
21. Russo, F., Lazzari, A.: Color Edge Detection in Presence of Gaussian Noise Using Nonlinear Prefiltering. *IEEE Trans. Instrum. Meas.* 54(1), 352–358 (2005)
22. Russo, F.: A Method Based on Piecewise Linear Models for Accurate Restoration of Images Corrupted by Gaussian Noise. *IEEE Trans. Instrum. Meas.* 55(6), 1935–1943 (2006)
23. Russo, F.: New Method for Performance Evaluation of Grayscale Image Denoising Filters. *IEEE Signal Processing Letters* 17(5), 417–420 (2010)

Application of Fuzzy Logic and Lukasiewicz Operators for Image Contrast Control

Angel Barriga and Nashaat Mohamed Hussein Hassan

Abstract. This chapter reviews image enhancement techniques. In particular the chapter is focused in soft computing technique to improve the contrast of images. There is a wide variety of contrast control techniques. However, most are not suitable for hardware implementation. A technique to control the contrast in images based on the application of Lukasiewicz algebra operators and fuzzy logic is described. In particular, the technique is based on the bounded-sum and the bounded-product. The selection of the control parameters is performed by a fuzzy system. An interesting feature when applying these operators is that it allows low cost hardware realizations (in terms of resources) and high processing speed.

1 Introduction

The sensory human systems are organized to respond rapidly to the temporary and spatial changes of the energy stimulus. When there is a temporary change in the energy applied to the sensor there is initially a strong response. Then the senses adapt rapidly (they respond less) to the constant and continued use of the energy. The visual system shows two types of behaviors: firstly to have the aptitude to answer (to see) both with weak lightings and with very brilliant lightings, and secondly to have the aptitude to discriminate between two objects that reflect very nearby intensities between them. To be able to adopt these types of behavior the visual system has two mechanisms: the mechanism of rapid adjustment and the mechanism of local adjustment. In the first case the retina changes its operative range (range of light intensity) approximately three tenths of second after the change taking place in the light intensity level. In case of the mechanism of local adjustment different parts of the retina adapt to different levels from lighting. The human eye is able to adapt to a range of illumination values around 10 orders of magnitude.

Angel Barriga · Nashaat Mohamed Hussein Hassan
Instituto de Microelectronica de Sevilla (CNM-CSIC)/University of Seville, Spain
e-mail: barriga@imse-cnm.csic.es

The luminance describes the energy of the stimulus and does not describe the changes of the energy. For this reason the contrast is defined in order to describe the changes of the energy. There are many proposals for the contrast measurement. Basically the contrast can be defined as the change of the relative luminance of the elements of an image. Therefore it corresponds to the difference of luminance that exists between two points of an image. The histogram of the image turns out to be a useful tool to determine the contrast in the image [1].

It is usual that the image captured by the sensor does not have the quality needed for the specific application that is required. This is due to deficient lighting conditions, aperture size, the shutter speed, noise coming principally from the capture sensor (quantization noise) and from the transmission of the image (fault on transmitting the information bits), etc. In these cases it is required the preprocessing of the image in order to improve its quality. One task of this preprocessing is to improve the contrast. As a result of the contrast enhancement there is visible additional information that apparently did not appear in the original image. This improves the image quality since it increases the dominance of some characteristics and reduces the ambiguity between different regions of the image.

In this chapter we are going to review the contrast adjustment techniques for image enhancement. These methods can be classified into two categories: spatial methods and methods in the frequency domain. The spatial methods are based in transforming the values of the pixels of the image whereas the methods in the frequency domain are based on modifying the Fourier transform.

Next, soft computing techniques to improve the contrast of images are described. Basically spatial techniques are discussed based on the uncertainty inherent in the image. Once the image is fuzzified, a transformation is performed in the fuzzy space. Some techniques apply some form of metrics to optimize results such as the fuzzy entropy measure. Finally, a transformation of the fuzzy space to the space of luminance levels is made.

Another aspect considered is related to the hardware implementations of contrast control systems. The hardware constraints in terms of used arithmetic (fixed point), width of words, processing resources, etc., impose limitations on the type of technique that can be implemented.

At the end of the chapter a technique based on the application of Lukasiewicz algebra operators will be described. In particular, the technique is based on the bounded-sum and the bounded-product in order to change the image contrast. These operators act as low pass filters as they produce a smoothing of the image and allow to eliminate noise . The application of Lukasiewicz algebra operators to an image produces a shift and an expansion of the histogram of the image The selection of the contrast control parameters is performed by a fuzzy system. An interesting feature when applying this technique is that it allows low cost hardware realizations (in terms of resources) and high processing speed.

2 Contrast Control Techniques

A definition of contrast is the peak-to-peak contrast or Michelson's contrast that measures the relation between the variation and the sum of two luminances. This definition is used in signal processing theory for determining the quality of a signal regarding to its noise level.

$$C_M = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \quad (1)$$

A further type of contrast measure is the variance. It is given by the following expression:

$$\sigma^2 = \frac{1}{MN} \sum_{k=1}^L (k - \bar{k})^2 n_k \quad (2)$$

where M and N are the dimension of the image, k is the value of the luminance in the range $[1, L]$, n_k is the frequency of the k luminance level, and \bar{k} is the mean value of the luminance distribution,

$$\bar{k} = \frac{1}{MN} \sum_{k=1}^L (kn_k) \quad (3)$$

When all the pixels have the same gray level its variance is zero, and when the difference between all the possible pairs of pixels is larger the variance is greater.

On the other hand the values $p_k = n_k/MN; k = 1, 2, \dots, L$ constitute a probability distribution on the set of the luminance values as $\sum_{k=1}^L (p_k)$. It is possible to use the entropy as a contrast measure [2]:

$$H = - \sum_{k=1}^L p_k \ln p_k \quad (4)$$

When the distribution of luminance tones of the pixels is uniform ($p_k = 1/L$) then the entropy reaches its maximum value (which is $\ln(L)$) which corresponds to an image with maximum contrast. This suggests that a standard measure in the interval $[0, 1]$ of the contrast of an image is $H/\ln(L)$.

Note that entropy is a measure of uncertainty. When it goes to zero corresponds to minimum contrast and for an image with uniform distribution, which corresponds to the maximum contrast, the uncertainty or lack of information is maximum.

Due to the process of digitalization of images the pixels are codified by a limited number of bits. For example, in the case of 8-bit monochrome images supposed to distinguish 256 levels of gray. If the range of variation in the brightness of the image is much smaller than the dynamic range of the camera then the true range of numbers will be much smaller than the full range from 0 to 255. That is, the image obtained at the output of the sensor s of the camera does not have to cover the full range. In many situations the recorded image will have a much smaller range of brightness values. These values can be found in the mid-range (intermediate gray values) or to bright or dark ends of the range.

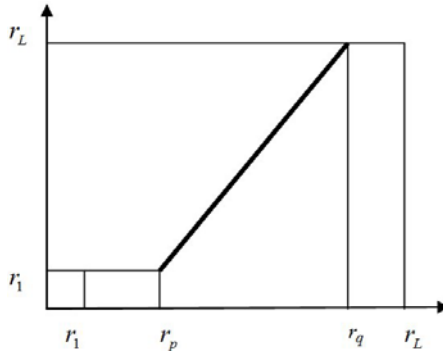


Fig. 1 Linear transformation

The visibility of the elements that form an image can be improved by stretching the contrast in order to reassign the values of pixels to fill the entire available range. This means that the pixels are interpolated between the extreme values of the dynamic range.

The contrast control techniques fall into two categories [3]: spatial techniques and techniques in the frequency domain. The first are based on making transformations of the values of the pixels of the image.

$$g(i, j) = T[f(i, j)] \quad (5)$$

where $f(i, j)$ and $g(i, j)$ represent the pixel (i, j) before and after applying the transformation defined by the function T .

The frequency domain techniques are based on the Fourier transform of the image. Thus the pixel value is calculated using the inverse Fourier transform:

$$g(i, j) = F^{-1}[H(u, v)F(u, v)] \quad (6)$$

where F and H are the Fourier transforms of $f(i, j)$ and a transformation $h(i, j)$ respectively.

A usual mechanism of contrast enhancement is to perform a linear interpolation [4] [5]. This technique of linear expansion of the contrast allows to increase the visual discrimination and is useful when the image has luminance variations that allow to distinguish between the elements that comprise it. A simple way is to apply the function shown in Figure 1. In this case the value of the new pixel is given by the following expression:

$$g(i, j) = a + b \times f(i, j) \quad (7)$$

where a and b values are calculated according to Figure 1. This process increases the image contrast by stretching the values of the luminance of the image to fill the full range.

It is possible to observe that if $f(i, j) = r_k$ then the new value of luminance of the pixel (i, j) , denoted by s_k , is determined by the following expression:

$$s_k = \frac{r_L - r_1}{r_q - r_p} (r_k - r_p) + r_1 \tag{8}$$

A type of transformation more general than the previous one to improve the contrast is:

$$f(x) = \begin{cases} \alpha \times u & \text{if } 0 \leq u \leq a \\ \beta \times (u - a) + v_a & \text{if } a \leq u < b \\ \gamma \times (u - b) + v_b & \text{if } b \leq u < L - 1 \end{cases} \tag{9}$$

Where u is the tone or luminance level of the original image and v is the luminance level of the transformed image.

With this transformation we made a change of the luminance value of the pixels in the image depending on the parameters α, β and γ . The transformation of Figure 2 enhances the contrast in the range $[0, a]$ since $\alpha > 1$, and also improves the contrast in the range $[b, L - 1]$ because $\gamma > 1$. Therefore this transformation improves the contrast of the darkest pixels, and also that of the lighter pixels.

One of the non-linear transformations most widely used is the Gaussian transformation that is given by:

$$g(i, j) = \frac{\phi(\frac{f(i,j)-0.5}{\sigma\sqrt{2}}) + [\frac{0.5}{\sigma\sqrt{2}}]}{\phi(\frac{0.5}{\sigma\sqrt{2}})} \tag{10}$$

where the brackets in the expression $[x]$ represent the floor function of x , and

$$\phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy \tag{11}$$

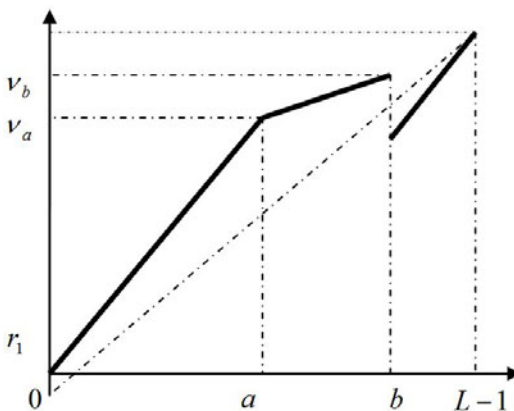


Fig. 2 Piecewise linear transformation

This transformation increases the contrast of the image making the dark parts darker and the bright part clearer.

3 Soft Computing Techniques

In [6] an excellent review of techniques for improving the contrast by applying fuzzy logic is presented. In this section we are going to check some of these techniques along with other recent proposals.

There are many strategies proposed for improving the contrast in images [7]. We will only describe a few of them (due to space limitations). Other techniques can be revised such as λ -enhancement [8] [9], fuzzy relaxation [10] [11], fuzzy morphology [12], fuzzy wavelet [13], and others.

3.1 Minimization of Image Fuzziness

One of the first applications of fuzzy logic to control the contrast was proposed in [14]. The technique is based in fuzzifying the image and modifying the pixels in the fuzzy domain. For this the INT operator (contrast intensification operator) is used. The degree of luminance of the pixel (i, j) in the interval $[0, 1]$ is denoted by μ_{ij} and is expressed as:

$$\mu_{ij} = G(f_{ij}) = [1 + \frac{f_{max} - f_{ij}}{F_d}]^{-F_c} \quad (12)$$

Where f_{max} is the maximum luminance of the image and f_{ij} is the luminance of pixel (i, j) . F_c and F_d are parameters of fuzzification. The operator INT is given by

$$INT(\mu_{ij}) = \begin{cases} 2\mu_{ij}^2 & \text{if } 0 < \mu_{ij} < 0.5 \\ 1 - 2(1 - \mu_{ij})^2 & \text{if } 0.5 < \mu_{ij} < 1 \end{cases} \quad (13)$$

This is a function that increases its value for μ_{ij} over the threshold 0.5 and decreases it below this threshold. This operator is applied iteratively so that the transformation that is performed at pixel (i, j) is given by:

$$\mu'_{ij} = INT_1(INT_r(\mu_{ij})) \quad r = 2, 3, \dots \quad (14)$$

The transformation of the fuzzy space to the space of luminance levels is achieved through the following expression:

$$f'_{ij} = G^{-1}(\mu'_{ij}) \quad (15)$$

where f'_{ij} is the luminance of pixel (i, j) for the new image and the function G^{-1} is the inverse of G .

Later, other authors have proposed modifications. Thus [15] considers the following fuzzification function:

$$\mu_{ij} = G(f_{ij}) = \left[1 + \frac{f_{ij} - f_{max}}{f_{max} - f_{min}}\right]^{-F_c} \tag{16}$$

In [16] [17] [18] successive generalizations of the operator of intensification are done, and a variable threshold is considered (instead of the constant value 0.5). This way the new intensification operator GINT (global contrast intensification) includes quality factors. The parameters of this operator are calculated optimizing the entropy of the image.

In [19] [20] the approach is based on the generalization of the fuzzy logic introduced by K. Atanassov in which the non-membership values are taken into account. This way they introduce the contrast intensification operator for Intuitionistic Fuzzy Sets (IFS).

3.2 Direct Method

In [21] local control is combined with edge detection and makes an adaptive control to the characteristics of the image. The degree of luminance of the pixel x_{ij} is expressed as:

$$\mu_{ij} = G(x_{ij}) = \begin{cases} 0 & \text{if } 0 \leq x_{ij} \leq a \\ \frac{(x_{ij}-a)^2}{(b-a)(c-a)} & \text{if } a \leq x_{ij} \leq b \\ 1 - \frac{(x_{ij}-a)^2}{(b-a)(c-a)} & \text{if } b \leq x_{ij} \leq c \\ 1 & \text{if } x_{ij} \geq c \end{cases} \tag{17}$$

where a, b, c are parameters of contrast control . Next a detection of edges is done by means of a gradient-based technique (Laplacian or Sobel operator) and the value of the edges is obtained in the fuzzy domain ($\delta_{\mu_{ij}}$). The next step is to calculate the mean edge value $E_{\mu_{ij}}$ for a window W_{ij} centered on the pixel x_{ij} .

$$E_{\mu_{ij}} = \frac{\sum_{ij \in W_{ij}} \mu_{ij} \delta_{\mu_{ij}}}{\sum_{ij \in W_{ij}} \delta_{\mu_{ij}}} \tag{18}$$

The contrast comes related to the membership value μ_{ij} is given as

$$C_{\mu_{ij}} = \frac{|\mu_{ij} - E_{\mu_{ij}}|}{|\mu_{ij} + E_{\mu_{ij}}|} \tag{19}$$

The contrast is modified by applying the modification constant σ_{ij} :

$$C'_{\mu_{ij}} = (C_{\mu_{ij}})^{\sigma_{ij}} \tag{20}$$

where $0 < \sigma_{ij} < 1$ to increase the contrast and $\sigma_{ij} > 1$ to reduce it. The calculation of this parameter is the critical step. For it the fuzzy entropy is applied. With this value of the contrast the modified membership value is calculated:

$$\mu'_{ij} = \begin{cases} \frac{E\mu_{ij}(1-C'\mu_{ij})}{1+C'\mu_{ij}} & \text{if } \mu_{ij} < E\mu_{ij} \\ \frac{E\mu_{ij}(1+C'\mu_{ij})}{1-C'\mu_{ij}} & \text{if } \mu_{ij} > E\mu_{ij} \end{cases} \quad (21)$$

The last step is the defuzzification. The grey level is calculated by means of the following expression:

$$x'_{ij} = \begin{cases} L_{min} & \text{if } \mu'_{ij} = 0 \\ L_{min} + \frac{L_{max}-L_{min}}{c-a} \sqrt{\mu'_{ij}(b-a)(c-a)} & \text{if } 0 < \mu'_{ij} < \frac{b-a}{c-a} \\ L_{min} + \frac{L_{max}-L_{min}}{c-a} (c-a - \sqrt{\mu'_{ij}(b-a)(c-a)}) & \text{if } \frac{b-a}{c-a} < \mu'_{ij} < 1 \\ L_{max} & \text{if } \mu'_{ij} = 1 \end{cases} \quad (22)$$

This technique allows an adaptive contrast enhancement since the modification parameter is calculated for each image.

3.3 Fuzzy Histogram Hiperbolation

Since the human perception of brightness is nonlinear, in [22] [23] is proposed to apply a hyperbolic transformation to the histogram given by

$$x'_{ij} = \frac{L-1}{e^{-1}-1} (e^{\mu(x_{ij})^\beta} - 1) \quad (23)$$

Where the membership function is described as

$$\mu(x_{ij}) = \frac{x_{ij} - x_{min}}{x_{max} - x_{min}} \quad (24)$$

The fuzzification parameter β determines the luminosity of the image. So if $\beta \rightarrow 0$ the image is brighter while if $\beta \rightarrow 1$ the image is darker.

3.4 Sharpening and Noise Reduction

F. Russo [24] uses a fuzzy network structure that combines contrast enhancement and noise removal. For it a structure of multiple outputs is used in order to improve the performance of the system since the method is recursive. Basically it is considered an $M \times N$ mask around pixel x_{ij} . The calculation of the output is given by:

$$\Delta x_{ij} = \frac{\alpha}{NM} \left[\sum_{m,n \in A} \mu_{R_1}(x_{ij}, x_{mn}, \alpha) - \sum_{m,n \in A} \mu_{R_2}(x_{ij}, x_{mn}, \alpha) \right] \quad (25)$$

Where R_q ($q = 1, 2$) represents the class of fuzzy relation described by:

$$\mu_{R_q}(u, v, \alpha) = \begin{cases} \max([1 - \frac{|u-v-\alpha|}{2\alpha}], 0) & \text{if } q = 1 \\ \max([1 - \frac{|u-v+\alpha|}{2\alpha}], 0) & \text{if } q = 2 \end{cases} \quad (26)$$

By means of parameter α ($0 < \alpha < L - 1$) different non linear behaviors are obtained. The output is obtained by:

$$y_{ij} = x_{ij} - \delta x_{ij} \quad (27)$$

In this case noise has been eliminated making a smoothing of the image. High values of α increase the cancellation of noise at the cost of increasing the blur. On the other hand a sharpening of the image can be realized if it is chosen $\alpha = \alpha_{max} = L - 1$ and the following operation is performed with the $\delta'x_{ij}$ value:

$$y_{ij} = x_{ij} + \delta'x_{ij} \quad (28)$$

The combination of both effects can be expressed as:

$$y_{ij} = x_{ij} \oplus (\Delta'x_{ij} - \Delta x_{ij}) \quad (29)$$

where operator $a \oplus b = \min(a + b, L - 1)$ is the bounded-sum .

The system architecture includes various fuzzy networks operating on different subsets of input data. The nonlinear behavior of the system is controlled by only one parameter α .

4 Hardware Realizations

There are hardware implementations circuits that perform the contrast control . In [25] a circuit implemented in a $0.25\mu m$ CMOS technology is described. The method is based on the following expression:

$$y_i = (x_i - L_{min})(M + US) \quad (30)$$

where $(M + US)$ is the value of the weight, US corresponds to a control parameter selected by the user and M is a weight that is calculated as follows:

$$M = \begin{cases} 1 + 2^{-n} & \text{if } MSB = 1 \\ 2^m + 2^{-n} & \text{in other case} \end{cases} \quad (31)$$

where MSB is the most significant bit of the difference with the value L_{min} , and the indices m and n are integer values and are calculated using specific heuristics. The system architecture consists of a module that performs the subtraction between the input pixel and the value L_{min} , the block that calculates the weight and the block which expands the range of the histogram formed by a shifter and adders. The circuit

has been synthesized from its VHDL description [26]. It has been implemented in a $0.25 \mu\text{m}$ CMOS technology. The resulting circuit has a cost of 2317 gates.

The method described in [4] applies in video images and is based on the piecewise linear functions approximation of the Cumulative Density Function (CDF). The CDF function is obtained by a sequential accumulation of the histogram. In this case the transformation of the image is made by means of linear interpolation:

$$F'(p_n) = \text{scalex}(F(p_n) - \frac{256n}{N}) + \frac{256n}{N} \quad (32)$$

where $F(p_n)$ and $F'(p_n)$ are the old and new values of pixel p_n , N is the total number of segments and n is the segment of pixel p_n .

The main problem is that the calculation of the CDF is costly in time so it is not recomputed between successive frames that contain similar images. The contrast control circuit consists of four modules. The CDF calculation module requires three comparators, three counters and memory to store the function. A median filter is used to alleviate the problem of abrupt changes in the image between consecutive frames. The authors give no implementation details of the circuit.

Other techniques are based on local transformations of the pixels and are called point operations. The point operations, or point to point functions, require in each step to know the value intensity of a single pixel, to which the desired transformation is applied. After processing the pixel is not needed, therefore these types of operation are called zero memory.

Point operations are performed more efficiently with lookup tables (LUTs). The LUTs are simple vector that use the value of the current pixel as an index of the vector. The new value is the vector element stored at that position. The new image is constructed by repeating the process for each pixel. Using LUTs avoids repeated and unnecessary computations. When working with images of, for example, 8 bits only need to calculate 256 values. In this case the size of the image is irrelevant since the value of each pixel of the image is a number between 0 and 255 and the result of lookup table produces another number between 0 and 255. These algorithms can be implemented without using any intermediate memory since the output image can be stored in the same memory space that the input.

5 Contrast Control by Means of Lukasiewicz Operators

The development of the theoretical concepts of the multi-valued logics began in the decade of the 20s by Jan Lukasiewicz, who established the generalization of the classic logic to the multi-valued logic. Later, at the end of the 50s, C.C. Chang formalized the multi-valued algebra based on Lukasiewicz logic. The basic operator's definitions are:

- bounded-sum : $x \oplus y = \min(1, x + y)$
- bounded-product : $x \otimes y = \max(0, x + y - 1)$

In order to visualize the meaning of the operators, Figure 3 shows the graphical representation of the bounded-sum and the bounded-product.

The application of the operators of Lukasiewicz in an image gives place to a transformation of the distribution of the levels of the pixels [27]. This transformation produces a shift from low levels to high or from high levels to low, ie after the application of Lukasiewicz operators, most of the gray levels of the image undergo a shift in the histogram.

The bounded-sum operator acts as a low-pass filter. This operator performs image smoothing and allows to reduce salt-peppers noise as well as Gaussian noise. Another effect of the bounded-sum operator is to perform a shift of the pixels to high levels. This way a clearer image is obtained. Figure 4 shows the effect of applying the bounded-sum to consecutive pixels of the original image. It is possible to observe the displacement of the pixels towards the white by giving a brightness image.

The contrast control using the bounded-sum can be done by introducing an additional parameter that allows to regulate the displacement of the frequency:

$$x \oplus y \oplus C \tag{33}$$

where C is the parameter of control of the contrast. The range of values that C (encoded with 8 bits) can take is in the interval $[-128,127]$. Figure 4 shows the effect of the bounded-sum with different values of the control parameter ($C = 0$ and $C = 30$).

The complementary operation to the bounded-sum corresponds to the bounded-product. This operator gives place to a displacement of the histogram towards the black. This effect is observed in Figure 5 that shows the result of applying the bounded-product and its histogram.

The control of the contrast applying the bounded-product it is realized by means of parameter C in the following expression:

$$x \otimes y \otimes C \tag{34}$$

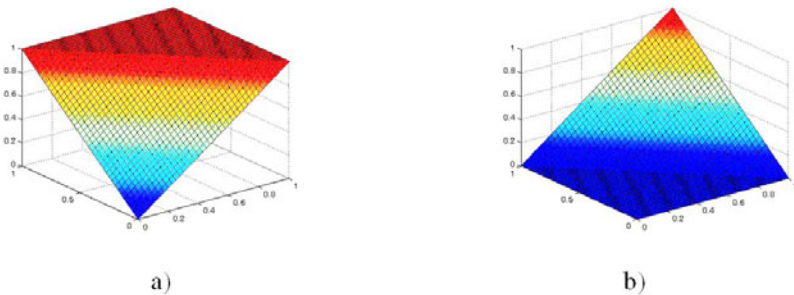


Fig. 3 Surfaces corresponding to the operators (a) bounded-sum and (b) bounded-product

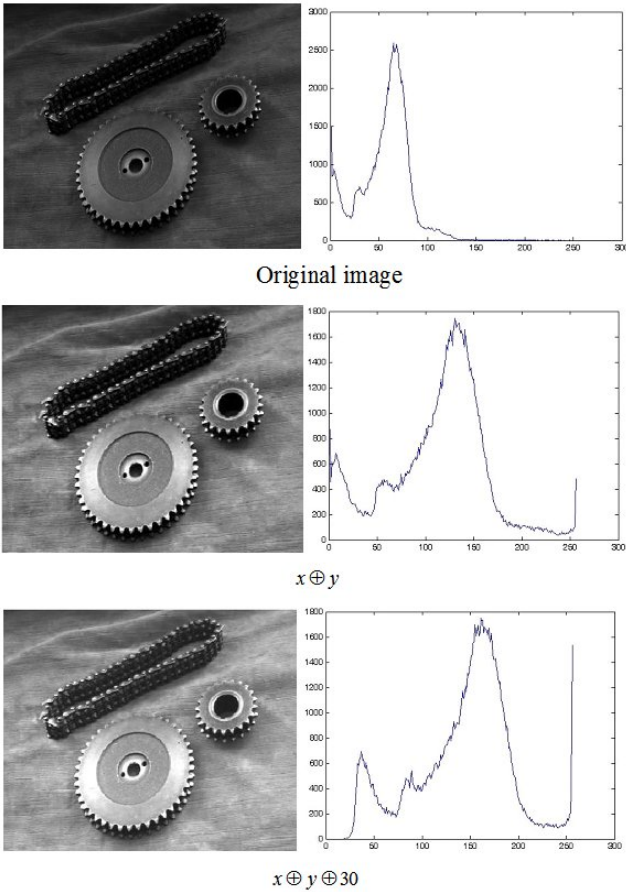


Fig. 4 Control of contrast using the bounded-sum and the histogram of the images

Figure 5 shows the application of the bounded-product with different values of the control parameter C .

6 Control Parameters Based on Fuzzy Logic

The technique of contrast control that has been presented is based on making a transformation of the histogram of the image by applying the operators bounded product and bounded sum. These operators give place to a shift and expansion of the values of the histogram. The control for this effect is achieved by a parameter C that allows to regulate the intensity of the transformation. The variation of contrast in an image needs not be uniform. So there may be regions where the contrast is lower than in other parts of the image. Therefore, the parameter C should adapt to

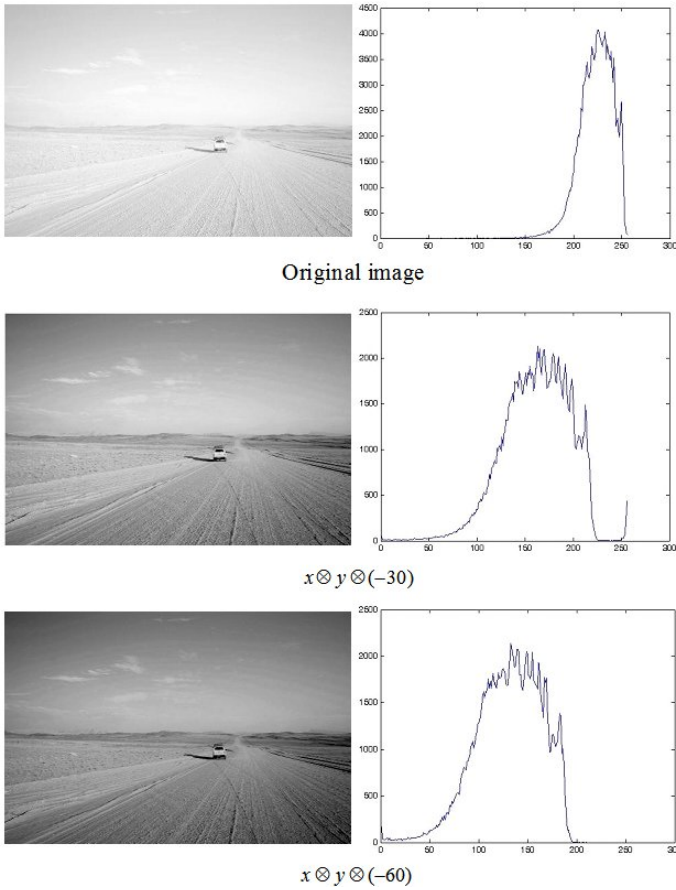


Fig. 5 Control of contrast using the bounded-product and the histogram of the images

each region of the image in order to improve the quality of the transformation. Thus the expression that regulates the contrast by means of the bounded sum is given by the following expression [28]:

$$x \oplus y \oplus f(x,y) \tag{35}$$

where x and y are pixels of the image and the parameter of control is the function $f(x,y)$.

The contrast control function $f(x,y)$ depends on the characteristics of each image and allows to adapt the control operation locally. In our case we have applied a heuristic based in a fuzzy logic inference mechanism. Thus the system of decision-making is based on criteria of proximity, that is, if the values of the pixels are very

close (low contrast) the function $f(x,y)$ must be high whereas if the pixels are far the function should be low.

- If x is Very Low and y is Very Low then $f(x,y)$ is F1
- If x is Very Low and y is Medium then $f(x,y)$ is F2
- If x is Very Low and y is Medium High then $f(x,y)$ is F3
- ...

Figure 6 shows the specifications of the fuzzy inference module that generate the function of control of contrast associated with the bounded sum. The membership functions correspond to seven equally spaced triangular functions with overlapping degree of two (VL Very Low, L Low, ML Medium Low, M Medium, MH Medium High, H High, VH Very High). The output of the system is composed by 9 singleton functions. The rule base details the heuristic described previously. Figure 7 shows the surface describing the behavior of the function of control of contrast.

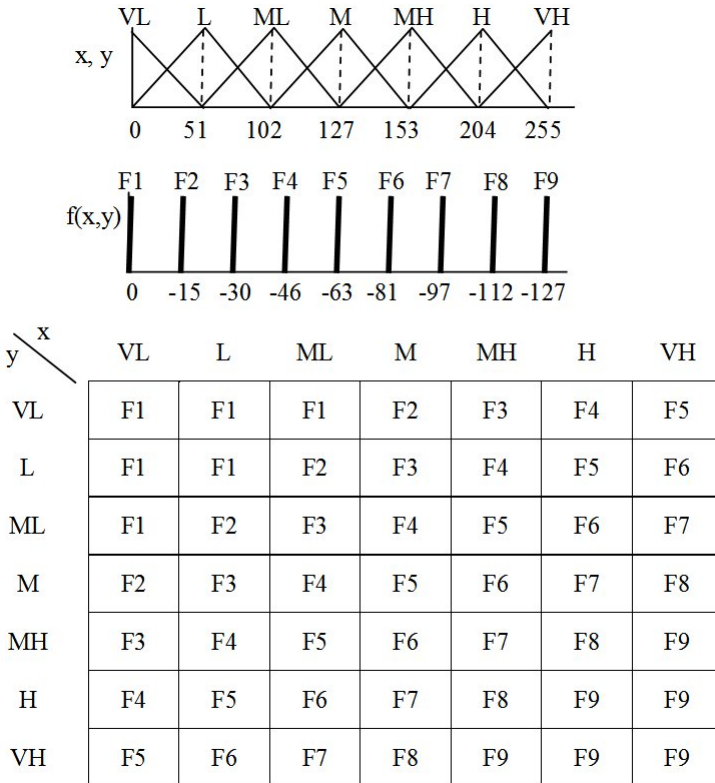


Fig. 6 Fuzzy inference module that generate the function of control of contrast associated with the bounded sum (7 membership functions for antecedents, 9 for consequent and 49 rules)

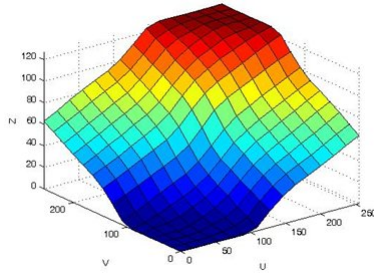


Fig. 7 Surface corresponding to the function of control of contrast)

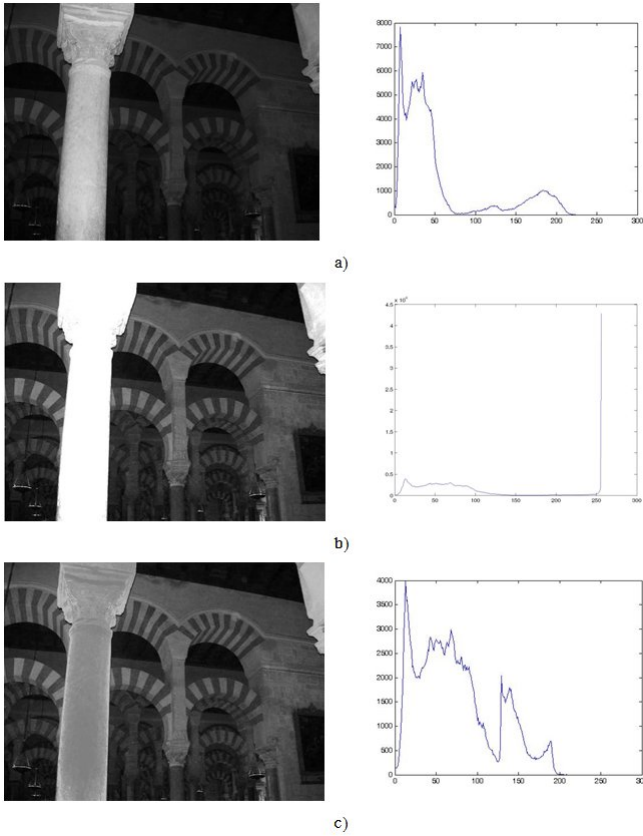


Fig. 8 a) Original image, b) $x \oplus y$, c) $x \oplus y \oplus f(x,y)$

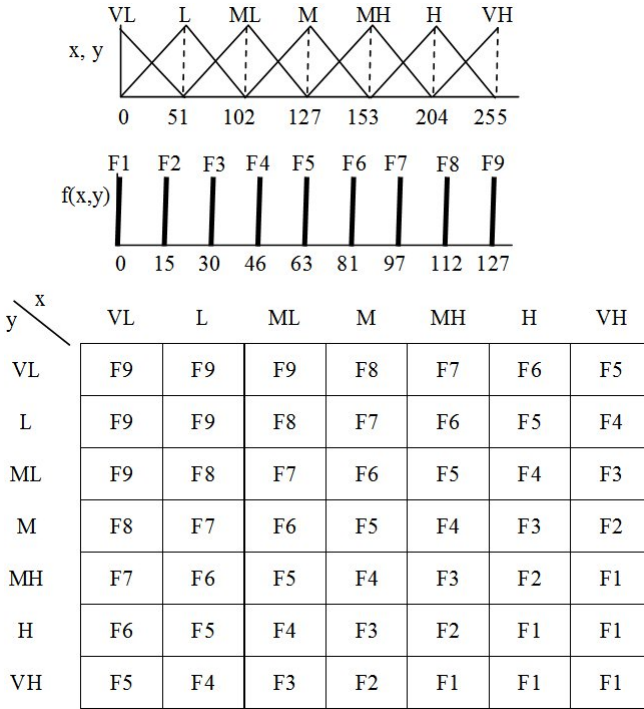


Fig. 9 Fuzzy inference module that generate the function of control of contrast associated with the bounded product (7 membership functions for antecedents, 9 for consequent and 49 rules)

Figure 8 shows an example of application of contrast control . The case of Figure 8b corresponds to the bounded sum; Figure 8c corresponds to the fuzzy control system. It can be observed in Figure 8 that in the zone corresponding to the column can be appreciated the effects of the control of contrast. It is noted that when control is not established the values of the column are saturated (they take the white value) so that contrast is reduced. Nevertheless when a local control is applied (case c) the contrast is improved in the zone of the column.

In the case of applying the bounded product the contrast is governed by the following expression:

$$x \otimes y \otimes f(x,y) \tag{36}$$

In the same way as in the case of the bounded sum the calculation of the function of the contrast control associated with the bounded-product is based on a fuzzy inference engine using the knowledge base shown in Figure 9.

The results obtained from the bounded product application are shown in Figure 10. Figure 10b shows the results of the bounded product without adaptation while Figure 10c corresponds to the control using the fuzzy system.

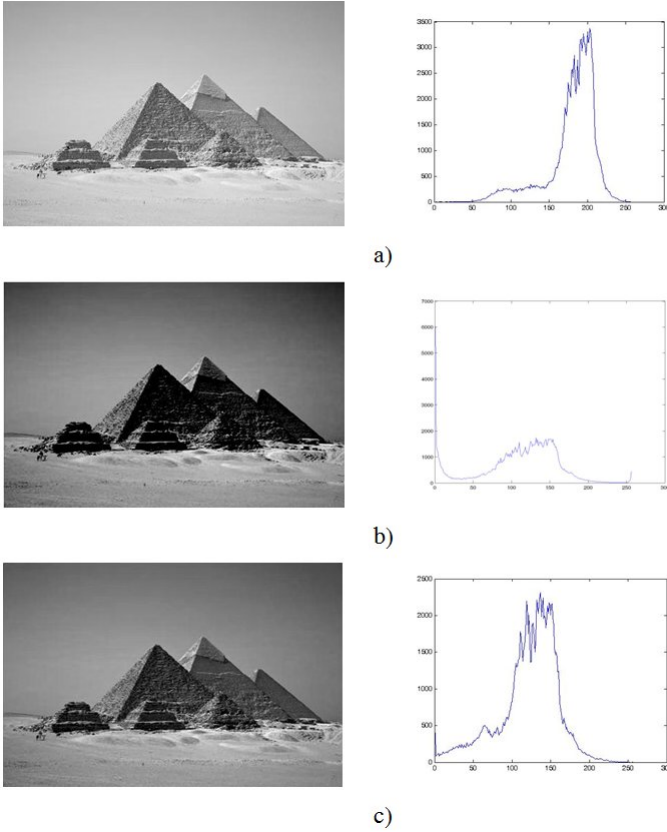


Fig. 10 a) Original image, b) $x \otimes y$, c) $x \otimes y \otimes f(x, y)$

7 Contrast Controller Architecture

The contrast control technique proposed previously applies each of the two operators (bounded sum and bounded product) depending on the characteristics of the image. This way the bounded sum is used in the case of dark images while the bounded product should be applied in clear images. However, in general, the images may have zones with different characteristics. This means that dark zones and clear zones can coexist in the same image. For that reason it is necessary to adapt the control mechanism to the local characteristics of the image. For it a decision-making system is required in order to determine the type of operator to be applied at each case (bounded sum in the dark area of the image and the bounded product in the clear area).

In agreement with this contrast control strategy the system is based in applying a mask that moves through the image. Depending on the local contrast the

system decides to apply the best operator. The global system is composed by 3 fuzzy inference engines as shown in Figure 11 [28]. The FIM1 and FIM2 fuzzy inference modules generate the functions of control of contrast associated with the bounded sum and the bounded product respectively. The FIM3 module corresponds to the decision-making system that selects the best operator. In this way the functionality of the system is given by the following expression:

$$x' = \begin{cases} x \oplus y \oplus f_1(x,y) & \text{if } z = Z1 \\ x & \text{if } z = Z2 \\ x \otimes y \otimes f_2(x,y) & \text{if } z = Z3 \end{cases} \quad (37)$$

The decision-making system is based on a fuzzy logic inference mechanism. The specification of the fuzzy system is shown in Figure 12. The membership functions are three functions equally distributed in the universe of discourse and with overlapping degree of two. On the other hand the membership functions of the consequent are 3 singletons ($Z1$, $Z2$ and $Z3$) that correspond to each of the three mechanisms to generate the output. Thus $Z1$ corresponds to perform the bounded sum while $Z3$ supposes to apply the bounded product. $Z2$ means that there is no change in contrast and therefore the output value corresponds to the input.

The rule base consists in 9 rules. When the contrast is low the bounded sum or the bounded product must be applied whereas if the contrast is high the output does not change with respect to the input.

Figure 13 shows an example of the application of the control system. The system receives the input image and calculates the images corresponding to the bounded sum and bounded product. The last stage corresponds to the selection of the output by means of a multiplexer controlled by the decision-making system FIM3. The output image is the result of a composition of the images generated by the bounded sum and the bounded product.

Figure 14 illustrates the block diagram of a fuzzy inference module (FIM). The main features of this architecture are low cost and high processing speed [29] [30].

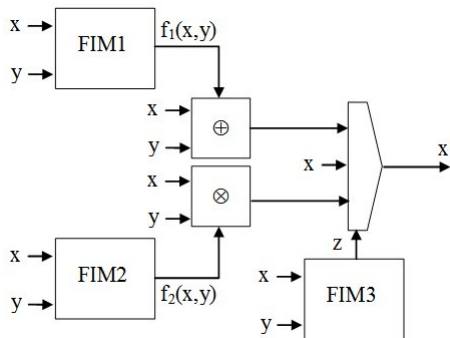


Fig. 11 Schematic of the system for control of contrast

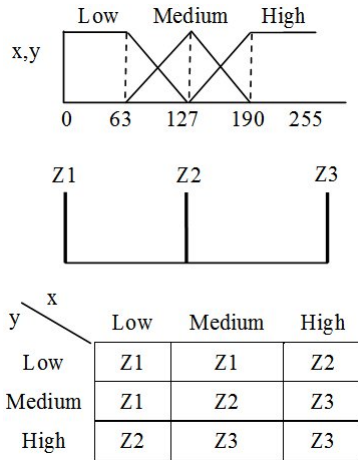


Fig. 12 Decision-making fuzzy system

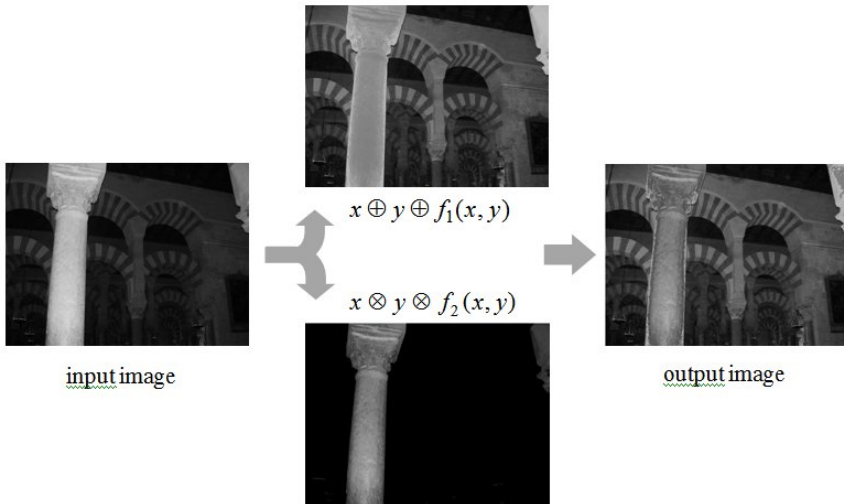


Fig. 13 Example of controller operation

The key elements to obtain this performance are due to limit the overlapping degree of the membership functions, the use of active rules inference mechanisms and the use of simplified defuzzification methods.

Membership function circuits (MFCs) at the fuzzifier stage calculate the degrees of membership for the inputs to the fuzzy sets which represent the antecedents of the

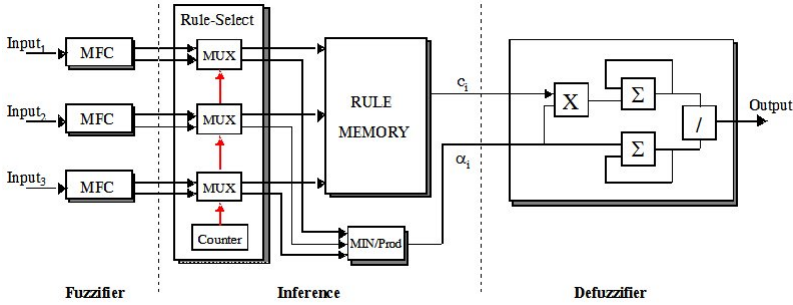


Fig. 14 Block diagram for active rule driven architecture of fuzzy inference module (FIM)

rules. Each MFC provides as many pairs of "label-activation degree" as overlapping degrees have been fixed for the system. The maximum number of active rules is limited by the overlapping degree. The MFC can be implemented using memories addressed by the inputs. The memory-based implementation has no restrictions on the shape of the membership functions, and the inference speed depends on the memory access time.

The inference stage is composed of an active-rule selection circuit (a counter-controlled multiplexer array is used for this purpose), a product circuit which evaluates the rule activation degree (α_i) by combining the antecedent activation degrees provided by the MFCs, and a rule memory that stores the parameters which define the rule consequents (c_i). Finally, the defuzzification stage computes the system output. The circuit in Figure 14 uses the Fuzzy Mean method according to the following equation:

$$Out = \frac{\sum_r \alpha_i c_i}{\sum_r \alpha_i} \quad (38)$$

This FIM module has been used in the design of the contrast control system. The contrast controller has been implemented on a low-cost FPGA device Spartan3 of Xilinx. The circuit has been optimized in order to reduce the cost in terms of resources. This circuit requires eight 16x16 bit and four 8x8 bit multipliers. On the other hand the knowledge base stored in memory. The memory requirements are: one 256x11 bit dual-port RAM, two 64x2 bit dual-port RAM and four 64x7 bit dual-port RAM. Regarding the processing speed the circuit has three stages of pipeline in order to improve the throughput. This allows to process HD images of 1920x1080 pixels at 60 frames per second.

8 Conclusions

In this chapter we have reviewed contrast control enhancement techniques focusing in fuzzy logic based methods. While there are many strategies proposed for improving the contrast in images very few of them are suitable for hardware

implementation. Taking into account this limitation a new contrast control technique has been described based on the bounded-sum and bounded-product operators. These operators produce a shift and expansion in the histogram of the image. The selection of the control parameters associated to both operators is performed by fuzzy systems. The resulting circuits are low cost with high processing speed. This makes the proposed contrast control system very suitable to be embedded together with the vision sensor and real time applications.

Acknowledgements. This work was supported in part by the European Community under the MOBY-DIC Project FP7-IST-248858 (www.mobydic-project.eu), by Spanish Ministerio de Ciencia y Tecnologia under the Project TEC2008-04920, and by Junta de Andalucia under the Project P08-TIC-03674.

References

1. Chen, Z.Y., Abidi, B.R., Page, D.L., Abidi, M.A.: Gray-Level Grouping (GLG): An Automatic Method for Optimized Image Contrast Enhancement-Part I: The Basic Method. *IEEE Transactions on Image Processing* 15(8), 2290–2302 (2006)
2. Khellaf, A., Beghdadi, A., Dupoisot, H.: Entropic Contrast Enhancement. *IEEE Transactions on Medical Imaging* 10(4), 589–592 (1991)
3. Gonzalez, R.C., Wintz, P.: *Digital Image Processing*. Addison-Wesley, Reading (1987)
4. Kim, S.Y., Han, D., Choi, S.J., Park, J.S.: Image Contrast Enhancement Based on the Piece-wise-Linear Approximation of CDF. *IEEE Transactions on Consumer Electronics* 45(3), 828–834 (1999)
5. Mantiuk, R., Daly, S., Kerofsky, L.: Display Adaptive Tone Mapping. *ACM Transactions on Graphics* 27(3), 68-1–68-10 (2008)
6. Tizhoosh, H.R.: Fuzzy image enhancement: an overview. In: Kerre, E.E., Nachttegael, M. (eds.) *Fuzzy Techniques in Image Processing*. Springer, Heidelberg (2000)
7. Haußecker, H., Tizhoosh, H.R.: Fuzzy Image Processing. In: *Handbook of Computer Vision and Applications*. Academic Press, London (1999)
8. Tizhoosh, H.R., Krell, G., Michaelis, B.: Enhancement: Contrast Adaptation Based on Optimization of Image Fuzziness. In: *Proceedings of IEEE International Conference on Fuzzy Systems FUZZ-IEEE 1998*, pp. 1548–1553 (1998)
9. Tizhoosh, H.R.: Adaptive -Enhancement: Type I versus Type II Fuzzy Implementation. In: *IEEE Symp. Series on Computational Intelligence* (2009)
10. Li, H., Yang, H.S.: Fast and Reliable Image Enhancement Using Fuzzy Relaxation Technique. *IEEE Transactions on Systems, Man and Cybernetics* 19(5), 1276–1281 (1989)
11. Zhou, S.M., Gan, Q.: A New Fuzzy Relaxation Algorithm for Image Contrast Enhancement. In: *International Symposium on Image and Signal Processing and Analysis*, pp. 11–16 (2003)
12. Wirth, M.A., Nikitenko, D.: Applications of Fuzzy Morphology to Contrast Enhancement. In: *Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS 2005*, pp. 355–360 (2005)
13. Liu, G.J., Huang, J.H., Tang, X.L., Liu, J.F.: A Novel Fuzzy Wavelet Approach to Contrast Enhancement. In: *International Conference on Machine Learning and Cybernetics*, pp. 4325–4330 (2004)
14. Pal, S.K., King, R.A.: Image enhancement using fuzzy set. *Electronic Letters* 16(10), 376–378 (1980)

15. Dong-liang, P., An-ke, X.: Degraded image enhancement with applications in robot vision. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pp. 1837–1842 (2005)
16. Hanmandlu, M., Jha, D., Sharma, R.: Color image enhancement by fuzzy intensification. In: International Conference on Pattern Recognition, vol. 3, pp. 310–313 (2000)
17. Hanmandlu, M., Jha, D.: An Optimal Fuzzy System for Color Image Enhancement. IEEE Transactions on Image Processing 15(10), 2956–2966 (2006)
18. Hanmandlu, M., Verma, O.P., Kumar, N.K., Kulkarni, M.: A Novel Optimal Fuzzy System for Color Image Enhancement Using Bacterial Foraging. IEEE Transactions on Instrumentation and Measurement 58(8), 2867–2879 (2009)
19. Vlachos, I.K., Sergiadis, G.D.: Intuistic Fuzzy Image Processing. In: Nachttegaal, M., Van der Weken, D., Kerre, E.E., Philips, W. (eds.) Soft Computing in Image Processing. Springer, Heidelberg (2007)
20. Palaniappan, N., Srinivasan, R.: Applications of intuitionistic fuzzy sets of root type in image processing. In: North American Fuzzy Information Society Annual Conference, NAFIPS (2009)
21. Cheng, H.D., Xu, H.J.: Fuzzy approach to contrast enhancement. In: International Conference on Pattern Recognition, vol. 2, pp. 1549–1551 (1998)
22. Tizhoosh, H.R.: Fuzzy image processing. Springer, Heidelberg (1997) (in German)
23. Tizhoosh, H.R., Krell, G., Lilienblum, T., Moore, C.J., Michaelis, B.: Enhancement and associative restoration of electronic portal images in radiotherapy. International Journal of Medical Informatics 49(2), 157–171 (1998)
24. Russo, F.: An image enhancement technique combining sharpening and noise reduction. IEEE Transactions on Instrumentation and Measurement 51(4), 824–828 (2002)
25. Kim, H.C., Kwon, B.H., Choi, M.R.: An Image Interpolator with Image Improvement for LCD Controller. IEEE Transactions on Consumer Electronics 47(2), 263–271 (2001)
26. Cho, H.H., Choi, C.H., Kwon, B.H., Choi, M.R.: A Design of Contrast Controller for Image Improvement of Multi-Gray Scale Image. In: IEEE Asia Pacific Conference on ASICs, pp. 131–133 (2000)
27. Hussein, N.M., Barriga, A.: Image Contrast Control based on?ukasiewicz’s Operators. In: IEEE International Symposium on Intelligent Signal Processing (WISP 2009), pp. 131–135 (2009)
28. Hussein, N.M., Barriga, A.: Image Contrast Control based on?ukasiewicz’s Operators and Fuzzy Logic. In: International Conference on Intelligent Systems Design and Applications, ISDA 2009 (2009)
29. Sánchez-Solano, S., Barriga, A., Jiménez, C.J., Huertas, J.L.: Design and Applications of Digital Fuzzy Controllers. In: Proceedings of IEEE International Conference on Fuzzy Systems FUZZ-IEEE 1997, pp. 869–874 (1997)
30. Baturone, I., Barriga, A., Sánchez-Solano, S., Jiménez, C.J., López, D.: Microelectronic Design of Fuzzy Logic-Based Systems. CRC Press, Boca Raton (2000)

Low Complexity Situational Models in Image Quality Improvement

Annamária R. Várkonyi-Kóczy

Abstract. Enhancement of noisy image data is a very challenging issue in many research and application areas. In the last few years, non-linear filters, feature extraction, high dynamic range (HDR) imaging methods based on soft computing models have been shown to be very effective in removing noise without destroying the useful information contained in the image data. Although, to distinguish among noise and useful information is not an easy task and may highly depend on the situation and aim of the processing. In this chapter new image processing techniques are introduced in the field of image quality improvement, thus contributing to the variety of advantageous possibilities to be applied. The main intentions of the presented algorithms are (1) to improve the quality of the image from the point of view of the aim of the processing, (2) to support the performance, and parallel with it (3) to decrease the complexity of further processing using the results of the image processing phase.

Keywords: image quality improvement, image enhancement, noise filtering, information extraction, situational models, anytime models, fuzzy decision making, complexity reduction.

1 Introduction

With the continued growth of multimedia and communication systems, the instrumentation and measurement fields have seen a steady increase in the focus on image data. Images contain measurement information of key interest for a variety of research and application areas such as astronomy, remote sensing, biology, medical sciences, particle physics, science of materials, etc. Developing tools and techniques

Annamária R. Várkonyi-Kóczy
Institute of Mechatronics and Vehicle Engineering
Óbuda University, Népszínház u. 8., H-1081 Budapest Hungary,
e-mail: varkonyi-koczy@uni-obuda.hu

to enhance the quality of image data plays, in any case, a very relevant role. In the last few years, non-linear filters, feature extraction, high dynamic range (HDR) imaging methods based on soft computing models have been shown to be very effective in removing noise without destroying the useful information contained in the image data. enhancement of noisy images, however, is not a trivial task. The filtering action should distinguish between unwanted noise (to be removed) and image details (to be preserved or possibly enhance). The main problem here is that ‘noise’ and ‘useful information’ are ill defined categories because they are usually dependent on the situation and on the intension and objective of the processing. What is characteristic or useful information in one application can be noise in another one. Soft computing, and especially fuzzy reasoning based methods are very well suited to model uncertainty and thus can effectively complete critical tasks where both noise cancellation and detail preservation (enhancement) have to be addressed.

In this chapter new, so-called situational models [10] of digital image processing are introduced (with special emphasis on complexity reduction, characteristic feature extraction, and useful information extraction), thus contributing to the variety of advantageous possibilities to be applied. The main intentions of the presented algorithms are (1) to improve the quality of the image from the point of view of the aim of the processing, (2) to support the performance, and parallel with it (3) to decrease the complexity of further processing using the results of the image processing phase.

In Section 2 corner detection is addressed. Corners (and edges) have an exceptional role in image processing related to pattern recognition and 3D reconstruction.

Section 3 deals with useful information extraction. “Useful” information means that the information is important from the further processing point of view. This information is enhanced and the, from this aspect non-important (in other situations possibly significant) image information is handled as noise, i.e. is filtered out.

Section 4 outlines a possible application area (automatic 3D reconstruction) permitted by the situational models of Sections 2–3 while Section 5 is devoted to the conclusions.

2 Corner Detection

Recently, the significance of feature extraction, e.g. corner detection has increased in computer vision, as well as in related fields. Corner detection helps to determine the shapes and the most characteristic points of an object and thus to reconstruct it. Corners are also useful in pattern recognition. In this section, a novel corner detection technique is presented, which is based on fuzzy reasoning and applies a special local structure matrix. Furthermore, by introducing a new attribute associated to the corners, the method efficiently supports further processing, e.g. point correspondence matching in stereo images or 3D reconstruction of schemes.

2.1 Overview

Corner detection plays an important role in computer vision [11], as well as in pattern recognition [20], in shape and motion analysis [19] and in 3D reconstruction [17] of a scene. The importance of corners can be underlined by many facts. Just to mention some of them: (1) Motion is unambiguous at a corner (and ambiguous along e.g. an edge), i.e. if we want to determine or follow some kind of motion it can be advised to analyze the movement of a corner point. (2) In most cases, shapes can approximately be reconstructed based on their corners [15]. (Of course, edges may have similar role however usually with higher complexity.) (3) 3D reconstruction from images has become an important common issue of several research domains. In recent time, the interest in 3D models has dramatically increased [2, 6].

More and more applications are using computer generated models. In many cases, models of existing scenes or objects are desired. Creating photorealistic 3D models of a scene from multiple photographs is a fundamental problem in computer vision and in image based modeling. The emphasis for most computer vision algorithms is on automatic reconstruction of the scene with little or no user interaction. The basic idea behind 3D model reconstruction, from a sequence of un-calibrated images, can be defined in more steps [17]: first, we need to relate the images in the whole sequence, then extract information based on pixel correspondences to be able to apply methods of epipolar geometry. The key element of this latter step is the correct pixel correspondence matching which also determines the reliability of the model reconstruction. Because of this, the pixels of interest have to be carefully chosen and as much support given as possible.

Real life image sequences contain many of the points that are better suited for automated matching than others. The environments of these points contain significant intensity variations and are therefore easy to differentiate from others. The correspondence between such points of interest has to be done by some kind of matching procedure. A possible approach to select points of interest is the corner detection. Corners in a scene are the end points of the edges. As we know, edges represent object boundaries and are very useful in 3D reconstruction of a scene. There are two important requirements for the feature of the points. First, points corresponding to the same scene point should be extracted consistently over the different views. Secondly, there should be enough information in the environment of the points so that the corresponding points can automatically be matched. If we take into consideration both aspects, i.e. feature type and area type attributes, based on the detected corners, it is possible to determine the corresponding points. If we are able to select points on the corners of objects we have a greater chance for matching the same corners in another image.

There are several known corner detection algorithms for the estimation of the corner points. These detectors are based on different principles, characteristic for the algorithm. It is known that there are corner detectors, whose functionality is based on a so-called local structure matrix, consisting of the first partial derivatives of the intensity function:

$$\mathbf{L}(x,y) = G(x,y) * \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} \quad (1)$$

where $G(x,y)$ corresponds to the Gaussian smoothing function and $*$ stands for the convolution operation. Examples of it are the Harris feature point detector [7] and Förstner's method [4].

Harris' method evaluates a comparison: the measure of the corner strength

$$R_H = \det(\mathbf{L}(x,y)) - k(\text{trace}(\mathbf{L}(x,y))) \quad (2)$$

is compared to an appropriately chosen constant threshold. If R_H exceeds the threshold, the point is taken as a corner. Here, $\text{trace}(\mathbf{L}(x,y)) = \lambda_1 + \lambda_2$, λ_1, λ_2 stand for the eigenvalues of matrix $\mathbf{L}(x,y)$, and k denotes a parameter effecting the sensitivity of the method (typical values for k are $k \in [0.04, 0.2]$).

Förstner determines the corners as the local maxima of function $H(x,y)$

$$H(x,y) = \frac{\det(\mathbf{L}(x,y))}{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (3)$$

Another well-known corner detector is the SUSAN (Smallest Univalued Segment Assimilating Nucleus) detector based on brightness comparison [14]. The algorithm does not depend on image derivatives, it uses the brightness values of the pixels. The first step of the algorithm is to place a circular mask around the pixel in question (the nucleus). After this, the method calculates the number of pixels within the circular mask which have similar brightness values to the nucleus. (These pixels define the so-called USAN.). The next step is to produce the corner strength image by subtracting the USAN size from a given geometric threshold. The possible false positives can be neglected by finding the USAN's centroid and its contiguity. The so called USAN area reaches a minimum (SUSAN), when the nucleus lies on a corner point. This method is more resistant to image noise than the previous ones.

The above, most well known algorithms all apply the following idea: When the calculated value of a certain feature (which is characteristic for a corner) exceeds a given, concrete threshold, the processed image point is usually detected as a corner. The effectiveness of these methods from corner detection point of view, is acceptable.

On the other hand, recent advances of the increased computer facilities and the new problems arising from the complexity of today's systems formulate new requirements for information processing. The previously accepted and used (classical) methods only partially cope with these challenges. As a consequence of the grown computational resources, more complex tasks can/are to be solved by more sophisticated solutions. The previous "processing" became "preprocessing" giving place to more advanced problem raising. Related to this, the aims of the newly developed preprocessing techniques have also changed. Besides the improvement of the performance of certain algorithms, the introduced methods have to accept and fulfill a new requirement, namely, to give more support to the "main" processing following

after. In image processing and computer vision it means that the previous processing tasks like noise smoothing, feature extraction (e.g. edge and corner detection), and even pattern recognition became part of the preprocessing phase and processing covers such fields like 3D modeling, medical diagnostics, and the automation of intelligent methods and systems (automatic 3D modeling, automatic analysis of ..., etc.).

The corner detection method presented in the following takes into consideration all of these requirements. Furthermore, its reliability outperforms that of the other methods and it opens a new possibility for automatic 3D modeling based on image point matching, partly because it analyzes both the area and feature type attributes and partly because it introduces a new parameter to be considered at the scene reconstruction.

2.2 *Detection of Corner Points*

Before starting to detect the corner points, it is advantageous to execute a preprocessing procedure including noise elimination and Gaussian smoothing. The cause of the first is trivial. Good performance noise elimination can be ensured by Russo's fuzzy filters (see details in [13] and also see his chapter in this book).

The need of the latter follows from the nature of digital representation. Digital images are stored as a collection of discrete pixels. An edge is represented as a series of points possibly resulting in small breaks in the edge which in many of the cases causes that false corners appear during the detection. In Fig. 1 the left hand side image illustrates how a line looks like (producing false corners) when the resolution of the image is finite. For improving the performance of the corner detection algorithm, the false corners should be eliminated before applying the corner detector by e.g. applying a Gaussian smoothing algorithm, which is usually used to "blur" images and to remove unimportant details and noise. In Fig. 1 the right hand side image shows how a line after smoothing appears in the image. After preprocessing the corner detection step can be performed.

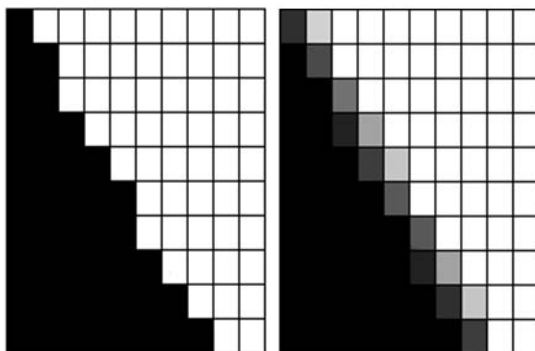


Fig. 1 Edge representation without smoothing (left) and after applying a smoothing algorithm (right)

Corners are local image features characterized by locations where the variations of the intensity function $I(x, y)$ are high in both the directions x and y , i.e. both partial derivatives I_x and I_y are large. This is the reason that many of the corner detection algorithms uses the local structure matrix $\mathbf{L}(x, y)$. The algorithm suggested here also utilizes $\mathbf{L}(x, y)$ as defined in eq. (1).

The local structure matrix $\mathbf{L}(x, y)$ can also be derived from locally approximating the autocovariance function of a real valued stochastic signal $I(x, y)$ (generated by a stochastic process) in the origin [3].

Based on $\mathbf{L}(x, y)$, the following beneficial function $H(x, y)$ can be defined according to Förstner's method [4]

$$H(x, y) = \frac{\left(\frac{\partial I}{\partial x}\right)^2 \left(\frac{\partial I}{\partial y}\right)^2 - \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y}\right)^2}{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (4)$$

While Förstner's algorithm compares $H(x, y)$ to a concrete threshold, a more effective corner detection method can be developed by handling $H(x, y)$ as a fuzzy term and by applying fuzzy reasoning during the decision making (whether a point is corner point or not). This is used for the characterization of the continuous transient between the localized and non-localized corner points, as well.

After preprocessing, the algorithm calculates the first derivatives of the intensity function $I(x, y)$ in each image point. For determining $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$, the following

convolution masks can be used: $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$, respectively.

It is also proposed to smooth each of the entries I_x^2 , I_y^2 , $I_x I_y$, in eq. (4). This step usually improves the effectiveness of the corner detection and it can be solved by applying again the Gaussian convolution kernel. As next, the values $H(x, y)$ are calculated for each image point based on the previously determined I_x^2 , I_y^2 , and $I_x I_y$ smoothed values. If the detected corners are neighbors, then we should keep only one as corner, the pixel having the largest calculated value $H(x, y)$. The others are to be ignored. By this, we can avoid multiple detection of a single corner.

In most of the cases, we can not unambiguously determine whether the analyzed image point is a corner or not based only on a certain concrete threshold value. Therefore, in the introduced algorithm fuzzy techniques are applied for the calculation of the values (corners) significantly increasing the rate of correct corner detection. As higher the calculated H value as higher the membership value, that the analyzed pixel is a corner. This approaches real situations where the view of a corner can vary between marked and dim.

After fuzzifying the H values into fuzzy sets and applying a fuzzy rulebase we can evaluate the "degree of cornerness" of the analyzed pixels. This attribute of the pixels can advantageously be used when searching for the corresponding corner points in stereo image pairs, as well. (point correspondence matching is an indefinite step of automatic 3D reconstruction. The consideration of an additional feature, i.e. the similarity of the degree of cornerness of the projections of a certain

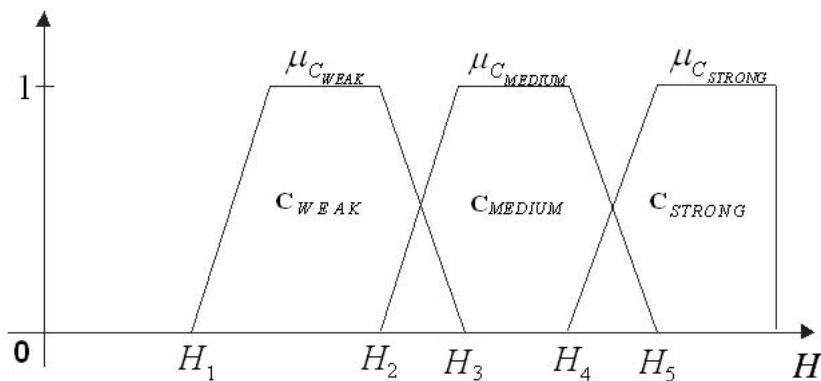


Fig. 2 Illustration of antecedent fuzzy sets C_{WEAK} , C_{MEDIUM} , and C_{STRONG} of universe H . The values H_k ($k=1,2,3,4,5$) serve for shaping membership functions $\mu_{C_{WEAK}}$, $\mu_{C_{MEDIUM}}$, $\mu_{C_{STRONG}}$ i.e. for tuning the sensitivity of the detector.

point in different pictures taken from near camera positions, can highly increase the reliability of the decision.

The antecedent and consequent fuzzy sets of the detector are illustrated in Figs. 2 and 3. In Fig. 2 (antecedent fuzzy sets) the horizontal axis represents the universe of the H values with three fuzzy sets defined, C_{WEAK} , C_{MEDIUM} , and C_{STRONG} corresponding to points being WEAK, MEDIUM, and STRONG corners, respectively. Parameters H_k ($k=1,2,3,4,5$) serve for the shaping of membership functions $\mu_{C_{WEAK}}$, $\mu_{C_{MEDIUM}}$, $\mu_{C_{STRONG}}$ by which the sensitivity of the described detector can be tuned.

In Fig. 3 (consequent fuzzy sets) the horizontal axis is the axis of universe I (output intensity) also with three fuzzy sets, I_{LOW} , I_{MEDIUM} , and I_{HIGH} . If the pixel is not at all a corner (none of the fuzzy rules are fired) then its intensity will be set

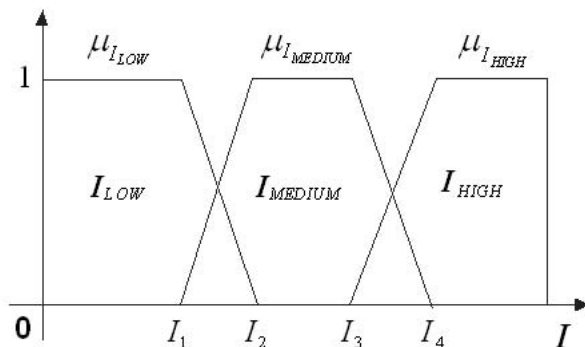


Fig. 3 Illustration of consequent fuzzy sets I_{LOW} , I_{MEDIUM} , and I_{HIGH} of universe I . The values I_k ($k=1,2,3,4,5$) serve for shaping membership functions $\mu_{I_{LOW}}$, $\mu_{I_{MEDIUM}}$, $\mu_{I_{HIGH}}$.

to zero, while in other cases the output intensity showing the degree of cornerness will be evaluated by the following fuzzy rulebase:

IF $(H(x,y), C_{WEAK})$ **THEN** $(I(x,y), I_{LOW})$,
IF $(H(x,y), C_{MEDIUM})$ **THEN** $(I(x,y), I_{MEDIUM})$,
IF $(H(x,y), C_{STRONG})$ **THEN** $(I(x,y), I_{HIGH})$,

which means that if the $H(x,y)$ value is member of the fuzzy set C_{WEAK} then the output intensity of the pixel is set to low, if the $H(x,y)$ value the member of the fuzzy set C_{MEDIUM} then the output intensity of the pixel is set to medium, etc. Let $\mu(\cdot)$ be the membership function of the consequent fuzzy set generated as the superposition of the rule consequents. As defuzzification algorithm, the center of gravity method is proposed, thus the intensity value of a pixel in the output image can be got by

$$I_o(x,y) = \frac{\sum_{i=1}^{I_{max}} \mu(I_i) I_i}{\sum_{i=1}^{I_{max}} \mu(I_i)} \quad (5)$$

where $I_o(x,y)$ denotes the intensity value of the pixel in the output image at position $[x,y]$ and I_{max} stands for the maximum of the intensity.

2.3 Experimental Results

While the new algorithm is no doubt advantageous from further processing and automation points of view, the author also investigated the effectiveness of the performance of the new method by comparing the rate of correct/false corner detections with that of the other methods. Appr. 35 different simulations have been made and the methods were tested by running them on different pictures partly taken from the literature (like the famous “Lena” photo) and partly chosen by the author as probably characteristic photos from corner detection point of view. For simplicity, gray scale images have been processed, with a maximum intensity of $L = 255$.

For ensuring the same conditions, during the preprocessing phase the same noise smoothing (typically a FIRE filter with $a=66$ and $b=100$, see [13]) has been applied on the images in each case. As an example of the results, we include here a “typical” running result in Table 1, containing the percentages of the correct/false/non detected corner points related to the total number of corners in the image. The parameters of the different corner detectors were set as follows:

- **fuzzy corner detector:** smoothing - by 2×2 Gaussian hump, fuzzy set - in the comparative runs, only one fuzzy set of cornerness is used with threshold $t=161$ and $\tan \alpha = 1/544$ (see Fig. 4) and further, the membership value corresponds to the strength of being corner. This simplification can be accepted here because the aim of the illustration is only to show the performance of the corner detection and not to use it for further complex processing, e.g automatic point correspondence

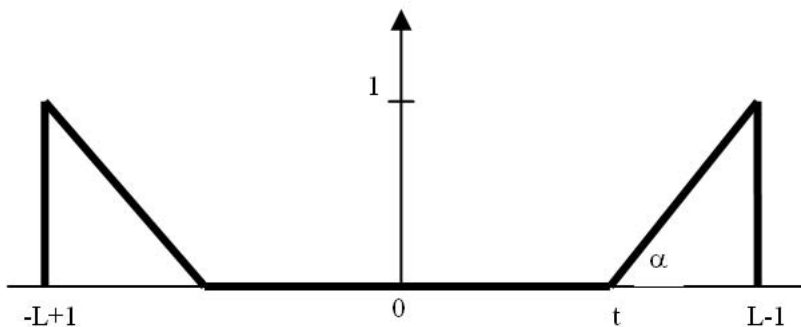


Fig. 4 Fuzzy set “corner”

matching. The threshold is set as in case of the Förstner’s method to make the comparison easier.

- **Förstner’s method:** threshold=161.
- **Harris corner detector:** $k=0.15$, threshold=5000.
- **SUSAN corner detector:** brightness threshold=10 (the maximum difference in gray levels between two pixels which allows them to be considered part of the same “region” in the image), geometric threshold=37 pixel fixed mask.

As illustration, we include two very simple examples to show the effectiveness of the new method. For more details and examples, see [15].

Fig. 5 presents results got using the fuzzy corner detector (a) by and (b) without image smoothing. This figure illustrates very well the improvement in the results when applying smoothing. (As curiosity, we would like to draw your attention to the following: In the right hand side image you can find a detected corner not located in a grid point which at first glance could be thought as false detection. However, at closer look we have found that during the manual cutting out of the check pattern we have made a corner by the scissors, i.e. the detection is correct.)

The next example serves for the comparison of different corner detection algorithms. Fig. 6 shows the fuzzy (noise) filtered photo of a part of a corridor with several lamps and doors. In Fig. 7 the corners detected by the introduced new fuzzy supported algorithm can be seen. By analyzing the results we can see that the most of the corners are detected and no false corner is found. Figs. 8-10 illustrate the results obtained by the Förstner’s, Harris, and SUSAN corner detection algorithms, respectively.

3 “Useful” Information Extraction

Nowadays, in digital image processing a large amount of research has been focused on information retrieval and image understanding. Typical examples are searching for similar objects/images in large databases and understanding the objects in pictures. The main point of these tasks is to determine and extract the most

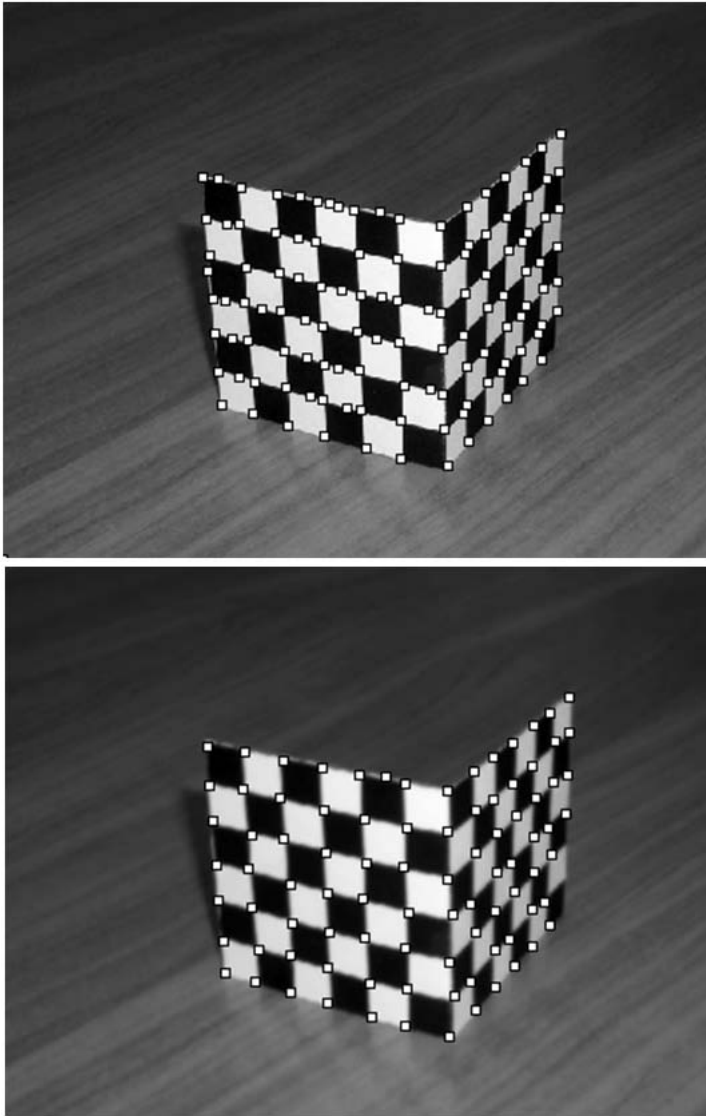


Fig. 5 Detected corners in the fuzzy filtered image (a) using fuzzy based detector without image smoothing, (b) using the same detector but with image smoothing

characteristic features of the objects in the images. Edges, corners, characteristic textures, etc. are typical examples, based on which objects can be identified and similarities can be found.

On the other hand, the growth of the database-sizes and also the increase in the complexity of the images, led to an information explosion which has become a



Fig. 6 Corridor - original photo

Table 1 Comparison of different corner detection methods

Methods / Corners	Correct[%]	False[%]	Not detected [%]
Fuzzy based corner detector (2×2 Gaussian hump, $a=100$, $b=161$, $\tan \beta = 1/544$)	84	0	16
Förstner's method (threshold=161)	78.8	0	21.3
SUSAN corner detector (brightness threshold=10, geometric threshold=37 pixel fixed mask)	52	4.7	48
Harris corner detector ($k=0.15$, threshold=5000)	71	15.3	29

serious limitation on the circle of effectively solvable tasks and also on the reliability of the results. Because of this, a huge expectancy has girded around any idea aiming to decrease the amount of processable information.

A possible approach leading to a solution can be the separation of the “significant” and “unimportant” parts of the characteristic features, i.e. the enhancement of

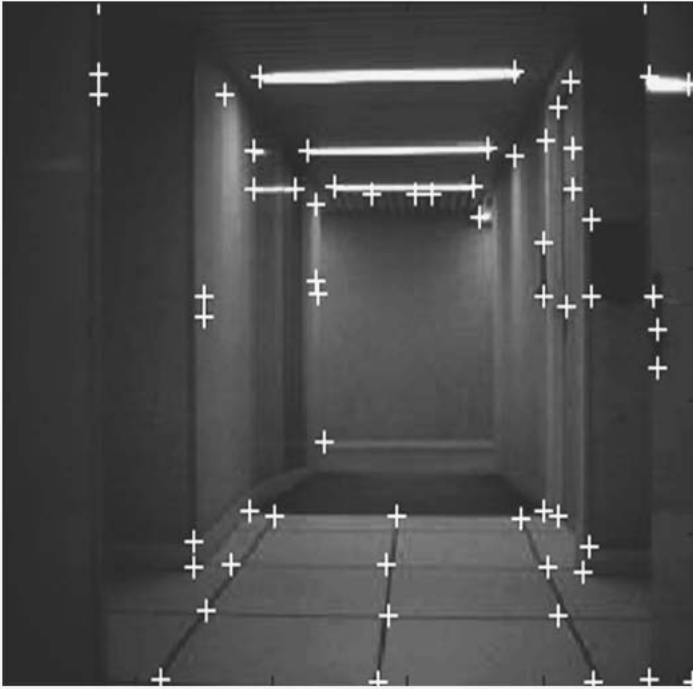


Fig. 7 Corridor - corner detection by the fuzzy corner detection algorithm

those features which carry primary information and to filter out the part, which represents information of minor importance. By this, the complexity of the searching and/or interpreting algorithms can be decreased while the performance increased.

Unfortunately, however traditional feature extraction methods (like corner and edge detectors) do not help too much in this latter task because their aim is to extract as many features as possible, regardless of their importance. Furthermore, “usefulness” is an ambiguous category and in many of the cases depends on the situation and aim of the processing.

This section describes a new edge processing method which can help in this matter. This useful information extraction method is able to extract the “primary” edges, i.e. the object boundaries which can advantageously be used in sketch based image retrieval algorithms.

3.1 Overview

The recent tremendous growth in computer technology parallel with the appearance of new advances in digital image processing has brought a substantial increase in the storage and aims of digital imagery. On one hand, this means the explosion of database-sizes, while on the other hand the increasing complexity of the stored

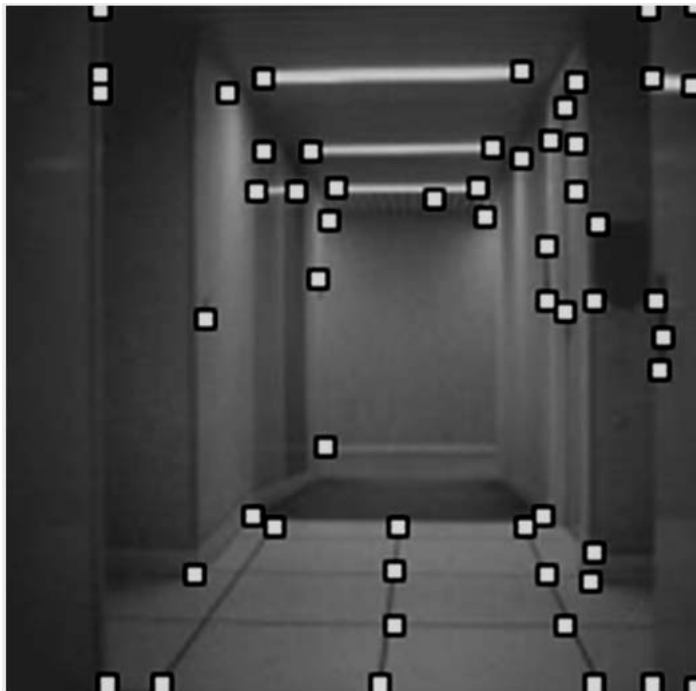


Fig. 8 Corridor - corner detection by Förstner's method

information calls for new, intelligent information managing methods. To address this challenge, a large number of the digital image processing algorithms which have been introduced in the past years apply soft computing and/or intelligent techniques. All of these algorithms aim some kind of (intelligent) feature extraction supporting the further, more advanced processing, like object recognition, image understanding, image information retrieval, etc. in a single photo or in (large) data bases.

A typical problem of the above type is searching for similar objects/images in large databases. Usually, this process is very time consuming, thus manual searching is not acceptable. A large amount of effort was put on the automation of the procedure. As a result, numerous methods of different kinds have been developed.

Some of the methods are based on the description of the images using text attributes, enabling the organization of images by topical or semantic hierarchies to facilitate easy navigation and browsing based on standard boolean queries [8, 1]. Automatically generating descriptive texts for a wide spectrum of images is not feasible, most text-based image retrieval systems require manual annotation of images. Obviously, annotating images manually is an expensive task for large image databases, and is often subjective, context-sensitive, and incomplete [8, 1]. Because of this reason, searching procedures based on imagecontent analysis have been



Fig. 9 Corridor - corner detection by the Harris corner detector ($k=0.001$)

developed, which can select the images more effectively as the text based retrieval methods.

The possibly most interesting and important step in image retrieval is the extraction of the “useful” features from the images. There are several characteristic attributes of the images (e.g. the edges and corners) which carry useful information and can be of help during the extraction of the primary information by appropriate techniques.

The edges in an image can advantageously be used when comparing two images and searching for similar objects. An image usually contains a lot of different edges, among which there are texture edges and object contour edges. From the point of view of image retrieval, only the latter ones are important because they carry the primary information about the shape of the objects. By considering both types of edges during the search/comparison, the complexity/ time need of the procedure might grow dramatically, and the (probably high number of) non-important details (edges) might lead to false decisions. As a consequence, it is of key importance to separate the “significant” and “unimportant” subsets of the edges, i.e. to enhance the ones which correspond to the object boundaries and thus carry primary information, but to filter out the others which represent information of minor importance.



Fig. 10 Corridor - corner detection by the SUSAN detector

The primary edge extraction method introduced in this section, applies surface deformation combined with fuzzy edge detection technique and leads to a solution of the above outlined problem.

3.2 *Surface Smoothing*

Let S_t be the surface describing an image to be processed, i.e. $S_t = \{(x, y, z); z = I(x, y, t)\}$, where variables x and y represent the horizontal and vertical coordinates of the pixels, z stands for the luminance value, which is the function of the pixel coordinates and of time t . Smoothing is performed by image surface deformation. Such a process preserves the main edges (contours) in the image. The surface deformation process satisfies the following differential equation [9]:

$$\frac{\partial I_t}{\partial t} = k\mathbf{n} \quad (6)$$

where k corresponds to the “speed” of the deformation along the normal direction \mathbf{n} of the surface S_t . In our case, value k is represented by the mean curvature of the

surface at location $[x, y]$, i.e. the speed of the deformation at a point is the function of the mean curvature at that point. The mean curvature is defined as

$$k = \frac{k_1 + k_2}{2} \quad (7)$$

where k_1 and k_2 stand for the principal curvatures. Starting from Eq. (7), the following partial differential equation can be derived (Because of the limitations on the volume, we skip the details of the deduction. For details, see [9]):

$$k = \frac{(1 + I_y^2) I_{xx} - 2I_x I_y I_{xy} + (1 + I_x^2) I_{yy}}{2(1 + I_x^2 + I_y^2)^{3/2}} \quad (8)$$

Here $I_x, I_y, I_{xx}, I_{xy}, I_{yy}$ stand for the partial derivatives with respect to the variables indicated as lower indices. Starting from Eq. (6) the surface at time $t + \Delta t$ (for small Δt) can be calculated as follows [9]:

$$I(x, y, t + \Delta t) = I(x, y, t) + k \sqrt{1 + I_x^2(x, y, t) + I_y^2(x, y, t)} \Delta t + o(\Delta t) \quad (9)$$

where $o(\Delta t)$ represents the error of the approximation.

Fig. 11 illustrates the virtual process of the surface deformation along the time.

3.3 Edge Detection

Fuzzy edge detection [13] is one of the key steps of the suggested primary edge extraction method. As in case of corner detection (see Subsection 2.2), we can not unambiguously determine whether an analyzed image point belongs to an edge or not based only on a certain concrete threshold value. The fuzzy interpretation of the intensity differences leads to much life-like results.

Let the pixel luminance of the original image at location $[x, y]$ be denoted by $z_{0,x,y}$. Considering the group of neighboring pixels which belong to a 3×3 window centered on $z_{0,x,y}$, the output of the edge detector is yielded by the following equation:

$$\begin{aligned} z_{x,y}^p &= (L - 1) \max \{m_{LA}(\Delta v_1), m_{LA}(\Delta v_2)\} \\ \Delta v_1 &= |z_{0,x-1,y} - z_{0,x,y}| \\ \Delta v_2 &= |z_{0,x,y-1} - z_{0,x,y}| \end{aligned} \quad (10)$$

where $z_{x,y}^p$ denotes the pixel luminance in the edge detected image and m_{LA} stands for the used membership function (see Fig. 12). $z_{0,x-1,y}$ and $z_{0,x,y-1}$ correspond to the luminance values of the left and upper neighbors of the processed pixel at location $[x, y]$. $L - 1$ equals to the maximum luminance value (e.g. 255).

For more details about fuzzy edge detection, see [13].

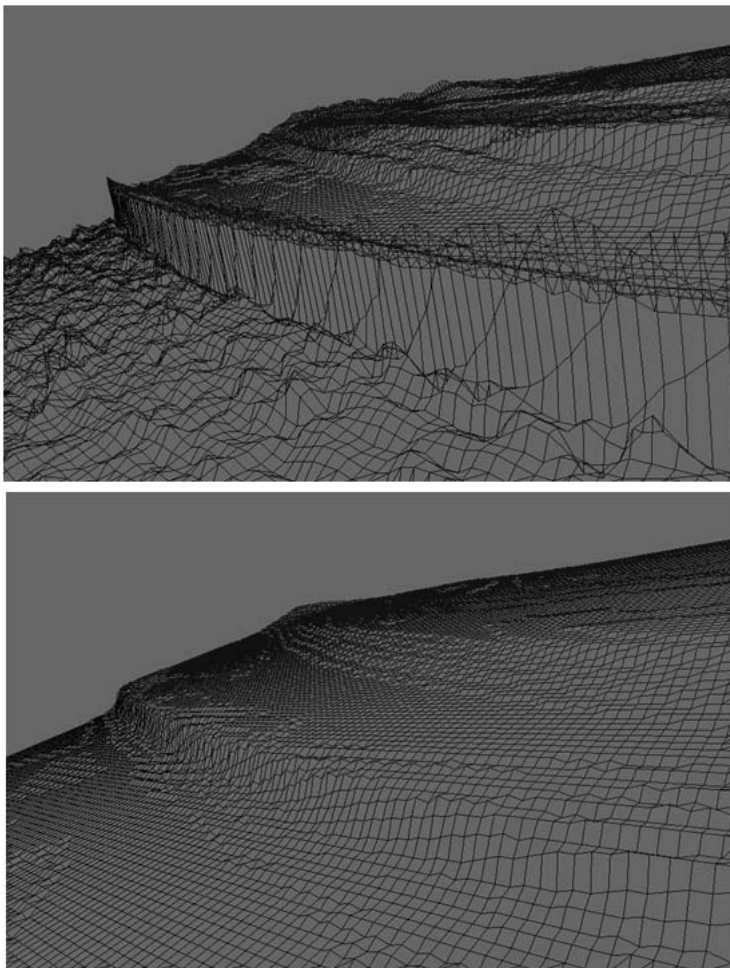


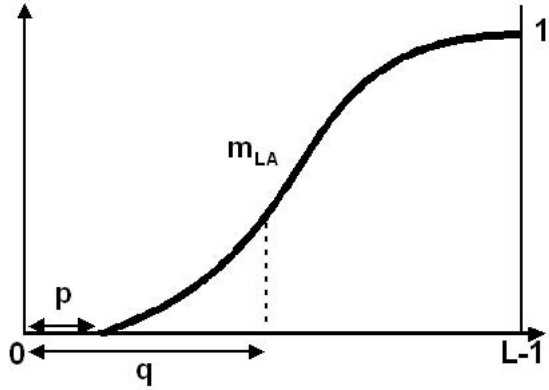
Fig. 11 Illustration of an image surface before (above) and after (below) the deformation

3.4 Edge Separation

The most characteristic edges of the objects are extracted in the smoothed image (Subsection 3.2 surface deformation) with the help of the constructed edge map of the original picture. The simultaneous analysis is performed for each edge point. The procedure is performed as follows:

For each edge point taken from the edge map of the original picture, the environment of the point is analyzed in the smoothed image. The analysis is realized by calculating the mean squared deviation of the color components (in case of greyscale images the grey-level component) in the environment of the selected edge point.

Fig. 12 Fuzzy membership function m_{LA} of “edge”. $L-1$ equals to the maximum intensity value, p and q are tuning parameters.



Let $\mathbf{p} = [p_x, p_y]$ be an edge point in the original image and let \mathbf{M} denote a rectangular environment of \mathbf{p} with width w and height h . The mean squared deviation is calculated as

$$d = \frac{\sum_{i=p_x-w/2}^{p_x+w/2} \sum_{j=p_y-h/2}^{p_y+h/2} (\mu - I(i, j, t_{stop}))^2}{hw} \quad (11)$$

where t_{stop} represents the duration of the surface deformation. In the case of grayscale images, μ denotes the average gray level inside the environment \mathbf{M} . For color images, the whole process should be done for each component separately and in this case μ corresponds to the average level of this color component inside the environment \mathbf{M} .

If the calculated deviation exceeds a predefined threshold value, then the edge point is considered as useful edge. Otherwise, the edge point is removed as unimportant. As result, an image containing only the most characteristic edges is obtained.

3.5 Illustrative Example

The effectivity of the primary information extraction method is illustrated by the analysis of a simple image. The aim of the processing is to extract the main object in the picture.

In Fig. 13 the photo of a car can be seen. (The original photo of the car is a color image and the analysis has been performed based on this picture). Fig. 14 shows the smoothed image using surface deformation. Figs. 15 and 16 represent the edge maps before and after the processing, respectively.

Fig. 16 illustrates very well that as a result of the processing, the complexity of the image significantly reduces, many of the (unimportant) details disappear and only the characteristic edges of the car are left. This helps filtering out the non-important details and enhancing the most significant features/objects in images. If



Fig. 13 Original image taken of a car



Fig. 14 Smoothed image using surface deformation based on mean curvature

we considered all of the possible edges during the searching/ object recognition/ 3D modeling, etc., it would cause that the complexity/ time need of the procedure might grow to a possibly intolerable degree and furthermore, the (probably high number of) non-important details (edges) might lead to false decisions and increased the uncertainty of the output (e.g. modeling) or caused that we disregarded recognizing an object. As a consequence, the separation of the significant and unimportant

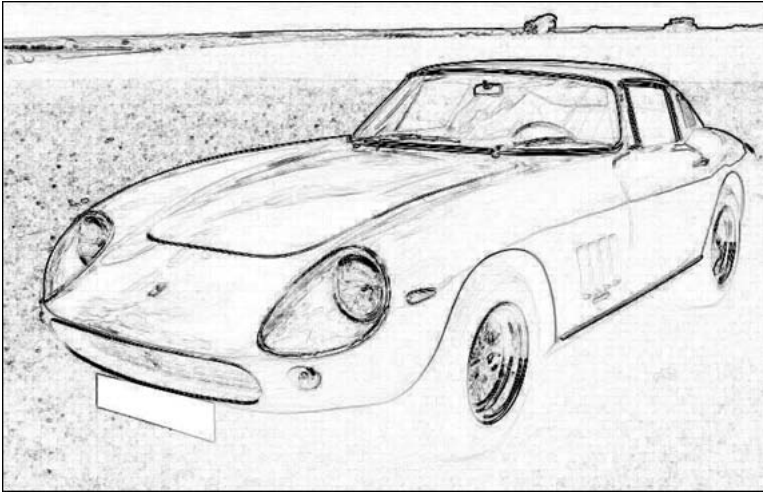


Fig. 15 Edge map of the original image

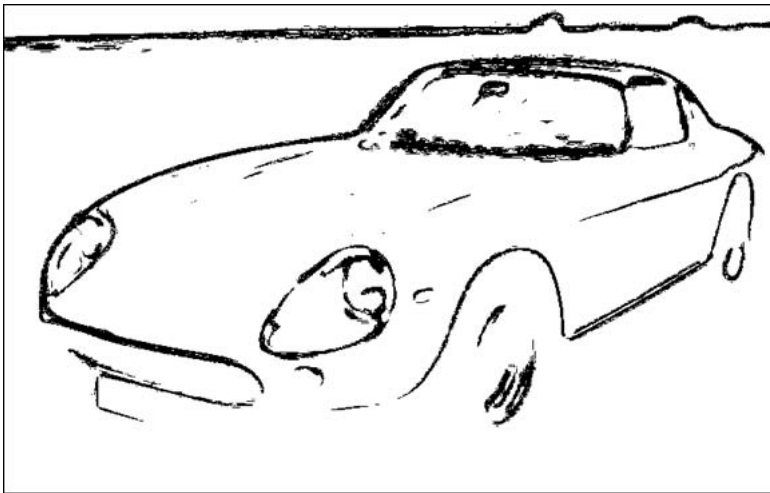


Fig. 16 Characteristic edges of the image extracted by the proposed useful information extraction method

subsets of the edges, i.e. the enhancement of those ones which correspond to the object boundaries and thus carry primary information and the filtering out of the others which represent information of minor importance, not only significantly decreases the computational complexity of the processing but is of key importance from interpretation point of view thus making easier image retrieval, object recognition, reconstruction of scenes, etc.

4 A Possible Application: 3D Reconstruction of Scenes

The topic of building 3D models from images is a relatively new research area in computer vision and, especially when the objects are irregular, not finished at all. In the field of computer vision, the main work is done at one hand on the automation of the reconstruction while on the other on the implementation of an intelligent human-like system, which is capable to extract relevant information from image data and not by all means on building a detailed and accurate 3D model like usually in photogrammetry is. For this purpose, i.e. to get the 3D model of scenes, to limit/delimit the objects in the picture from each other is of key importance [12].

The basic concept of the 3D model estimation based on 2D images can be summarized as follows: As the first step, the pictures, used in the 3D-object reconstruction are preprocessed, which starts with noise elimination and edge detection by applying the fuzzy filters and fuzzy edge detection algorithm described in [13]. This is usually followed by the primary edge extraction method (see Section 3).

For the modeling the determination of the primary edges and corners are very important because they carry the most characteristic information about the shape of the objects to be modeled. The applied corner detection method utilizes the notion that a corner is indicated by two strong edges. It also applies fuzzy reasoning and the used local structure matrix composed of the partial derivatives of the color (gray level) intensity of the pixels is extended by fuzzy decision making. The algorithm assigns also a new attribute, the fuzzy measure of being a corner, to the analyzed pixel. This property of the corners can advantageously be used at the searching for the corresponding corner points in stereo image pairs (Section 2).

The next step is the determination of the 3D coordinates of the extracted edge points. First the corner point correspondences are determined which is followed by the determination of the edge correspondences in the different images. If the angle between the camera positions is relatively small then after the estimation of the projection matrices of the images, necessary for the calibration, (they map the projective space to the cameras' retinal plane: 3D to 2D, see [12]) the corresponding points can be calculated automatically with high reliability in each image. We search for the characteristic corner or edge points lying (in fuzzy sense) on the epipolar line and then the point correspondence matching is done by minimizing the fuzzy measure of the differences of the environment of the points with the help of a fuzzy supported searching algorithm [16]. The similarity of the above mentioned cornerness is also considered. (The corresponding corner points keep their cornerness property in the pictures near to each other with high reliability). Having the point correspondences we can calculate the 3D position of the image points (the camera calibration is solved by the determination of the Perspective Projection Matrix [12]) and in the knowledge of the 3D coordinates and the correspondences of the significant points the spatial model of the scene can easily be built.

The situational models applied in 3D model building open a way for the total automation (carried out without any human intervention) with high reliability (because of the fuzzy point matching algorithm based on fuzzy corner detection and

fuzzy minimization of the environmental differences) and low complexity (due to the application of the primary information enhancement technique).

5 Conclusions

Situational models has been designed for modeling complex situations where the traditional cybernetics models haven't proved to be enough sufficient because the characterization of the system is situation dependent, incomplete or ambiguous, containing unique, dynamically changing, and unforeseen situations. In image processing, "information" and "noise" are such categories which can not unambiguously distinguished because their definition may highly depend on the situation and the aim of the processing.

In this chapter, situational models of image processing techniques (corner detection and primary information extraction) are introduced. The presented algorithms aim to improve the quality of the images from the point of view of further processing, to support the performance, and parallel with it to decrease the complexity of the processing. The adaptivity of the models makes possible to reduce and solve complex problems where up till recently, the complexity of the processing has limited the effective realization.

The methods presented in the chapter open new possibilities in automation and intelligent processing. They can advantageous be used in many areas of 2D and 3D applications, in robotics, computer vision, sketch based image retrieval methods, intelligent monitoring and analysis systems, vehicle system dynamics, etc. As example, a possible application area, 3D reconstruction of scenes is also presented briefly.

Acknowledgments. This work was sponsored by the Hungarian Fund for Scientific Research (OTKA K 78576) and the Structural Fund for Supporting Innovation in New Knowledge and Technology Intensive Micro- and Spin-off Enterprises (GVOP-3.3.1-05/1.2005-05-0160/3.0).

References

1. Assfalg, J., Bimbo, A.D., Pala, P.: Using multiple examples for content-based retrieval. In: Proc. of the Multimedia and Expo, ICME 2000, vol. 1, pp. 335–338 (2000)
2. Debevec, P.E., Taylor, C.J., Malik, J.: Modeling and rendering architecture from photographs a hybrid geometry and image based approach. In: ISIGGRAPH (1996)
3. Felsberg, M.: Low-Level Image Processing with the Structure Multivector, PhD thesis, Inst. of Computer Science and Applied Mathematics. Christian-Albrechts-University of Kiel (2002)
4. Förstner, W.: A feature based correspondence algorithm for image matching. *Int. Arch. Photogramm. Remote Sensing* 26, 150–166 (1986)
5. Gray, A.: The Gaussian and mean curvatures and Surfaces of Constant Gaussian Curvature. In: *Modern Differential Geometry of Curves and Surfaces with Mathematica*, 2nd edn., ch. 21, 16.5, pp. 373–380, 481–500. CRC Press, Boca Raton (1997)

6. Grossmann, E., Ortin, D., Santos-Victor, J.: Single and multi-view reconstruction of structured scenes. In: Proc. of the 5th Asian Conf. on Computer Vision, Melbourne, Australia (2002)
7. Harris, C. and Stephens, M.: A combined corner and edge detector. In: Proc. of the 4th Alvey Vision Conf., pp. 189-192 (1988)
8. Long, F., Zhang, H., Dagan Feng, D.: Fundamentals of Content Based Image Retrieval. In: Multimedia Information Retrieval and Management Technological Fundamentals and Applications, pp. 1–26. Springer, Heidelberg (2003)
9. Lu, C., Cao, Y., Mumford, D.: Surface Evolution under Curvature Flows. *Journal of Visual Communication and Image Representation* 13, 65–81 (2002)
10. Madarász, L., Andoga, R., Fözö, L., Lázár, T.: Situational control, modeling and diagnostics of large scale systems. In: Rudas, I.J., Fodor, J., Kacprzyk, J. (eds.) *Towards Intelligent Engineering and Information Technology*. SCI, vol. 243, pp. 153–164. Springer, Heidelberg (2009)
11. Paragios, N., Chen, Y., Faugeras, O.: *Handbook of Mathematical Models in Computer Vision*. Springer, Heidelberg (2005)
12. Pollefeys, M.: *Self-Calibration and Metric 3D reconstruction from Uncalibrated Image Sequences*. PhD thesis, ESAT-PSI, K.U. Leuven (1999)
13. Russo, F.: Recent Advances in Fuzzy Techniques for Image Enhancement. *IEEE Transactions on Instrumentation and Measurement* 47(6), 1428–1434 (1998)
14. Smith, S.M., Brady, M.: SUSAN - a new approach to low level image processing. *Int. Journ. of Computer Vision* 23(1), 45–78 (1997)
15. Várkonyi-Kóczy, A.R.: Fuzzy Logic Supported Corner Detection. *Journal of Intelligent and Fuzzy Systems* 19(3), 41–50 (2008)
16. Várkonyi-Kóczy, A.R., Rövid, A.: Soft Computing Based Point Corresponding Matching for Automatic 3D reconstruction. *Acta Polytechnica Hungarica (Special Issue on Computational Intelligence)* 2(1), 33–44 (2005)
17. Várkonyi-Kóczy, A.R., Rövid, A., Ruano, M.G.: Soft Computing Based Car Body Deformation and EES Determination for Car Crash Analysis Systems. *IEEE Trans. on Instrumentation and Measurement* 55(6), 2300–2308 (2006)
18. Várkonyi-Kóczy, A.R., Rövid, A.: Fuzzy Logic Supported primary edge extraction in image understanding. In: CD-ROM Proc. of the 17th IEEE Int. Conference on Fuzzy Systems, FUZZ-IEEE 2008, Hong Kong, China (2008)
19. Velastin, S.A., Yin, J.H., Davies, A.C., Vicencio-Silva, M.A., Allsop, R.E., Penn, A.: Automated Measurement of Crowd Density and Motion using Image Processing. In: Proc. of the 7th IEE Int. Conf. on Road Traffic Monitoring and Control, London, UK, pp. 127–132 (1994)
20. Xie, X., Sudhakar, R., Zhuang, H.: Corner detection by a cost minimization approach. *Pattern Recognition* 26(8), 1235–1243 (1993)

A Flexible Representation and Invertible Transformations for Images on Quantum Computers

Phuc Q. Le, Abdullahi M. Iliyasu, Fangyan Dong, and Kaoru Hirota

Abstract. A flexible representation for quantum images (FRQI) is proposed to provide a representation for images on quantum computers which captures information about colors and their corresponding positions in the images. A constructive polynomial preparation for the FRQI state from an initial state, an algorithm for quantum image compression (QIC), and invertible processing operations for quantum images are combined to build the whole process for quantum image processing based on FRQI. The simulation experiments on FRQI include storage and retrieval of images and detecting a line from binary images by applying quantum Fourier transform as a processing operation. The compression ratios of QIC between groups of same color positions range from 68.75% to 90.63% on single digit images and 6.67% to 31.62% on the Lena image. The FRQI provides a foundation not only to express images but also to explore theoretical and practical aspects of image processing on quantum computers.

1 Introduction

In recent years quantum computation and quantum information have generated so much interest especially with the prospect of employing its insights to empower our knowledge on information processing. After Feynman's proposal of quantum computers [6], Shor [15] discovered a quantum algorithm to factor integer numbers in polynomial time in 1994. This was closely followed by Grover's quadratic speed-up database search algorithm [8] on the quantum computation model. These results and the unavoidably inefficient simulation of quantum physics on classical computers

Phuc Q. Le · Abdullahi M. Iliyasu · Fangyan Dong · Kaoru Hirota
Department of Computational Intelligence and Systems Science,
Interdisciplinary Graduate School of Science and Engineering,
Tokyo Institute of Technology,
G3-49, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan
e-mail: phuc1q@hrt.dis.titech.ac.jp

[6] provide the solid evidence of the strength of quantum computers over classical ones. Quantum computation has appeared in various areas of computer science such as information theory, cryptography, image processing [14] because there are inefficient tasks on classical computers that can be overcome by exploiting the power of the quantum computation. Processing and analysis of images in particular and visual information in general on classical computers have been studied extensively. On quantum computers, the research on images has faced fundamental difficulties because the field is still in its infancy. To start with, what are quantum images or how do we represent images on quantum computers? Secondly, what should we do to prepare and process the quantum images on quantum computers?

One of the most active fields in quantum computation and information is quantum image processing. Quantum signal processing transformations such as Fourier[14], wavelet[7], and discrete cosine[9, 16] are proven to be more efficient than their classical versions. Using these efficient operations for image processing applications previously inefficient approaches involving classical operations are realizable[2]. Parallelism in quantum computation can speed up many image processing tasks which have characteristics of parallelism[14]. Some concepts of quantum images have been proposed like Qubit Lattice[17, 18] and Real Ket[10] in order to make the connection between quantum algorithms and image processing applications. Some impossible processing operations on quantum computers[11] indicate the fundamental difference between quantum and classical operations. The complexity of the preparation of quantum images and the application of quantum transforms to process quantum images, however, have not been studied.

In this research, a flexible representation of quantum images (FRQI) which captures information about colors and their corresponding positions in an image into a normalized quantum state is proposed. After the proposal of FRQI, the computational and image processing aspects on FRQI are studied:

- The complexity (the number of simple operations) of the preparation for FRQI,
- The method to reduce number of simple operations that are used in the FRQI preparation step or quantum image compression (QIC),
- Three types of invertible image processing operators on FRQI.

The first stage of any image processing task based on the FRQI representation involves the use of Hadamard and controlled rotation operations in order to prepare the input image. As proven by the Polynomial Preparation theorem, the total number of simple operations used in the preparation is polynomial for the number of qubits which are used to encode all positions in an image. Human vision is incapable of effectively distinguishing slight variations in color, the QIC algorithm based on FRQI reduces the computation involving the same color positions by integrating controlled part of the controlled rotations in the groups. To further process images encoded in FRQI representation, processing operators based on unitary transformations to operate on the colors only, colors at some positions and the combination of both colors and positions are proposed. Experiments which involve classical simulation of the FRQI state and processing operations are performed to confirm the capacity of FRQI on storage and retrieval quantum images, compression ratio among the same color

groups on QIC algorithm and an application of an image processing operator for the third type, using quantum Fourier transform, for a line detection in binary images on quantum computers.

The results indicate that the FRQI can be the basis to represent and process quantum images. The preparation and processing operations on FRQI improve the whole procedure of quantum image processing, i.e. a quantum computer starts from its initial state, then it is prepared to the FRQI state, and it is finally transformed by processing operations. The QIC algorithm suggests a way to reduce the main resources used to represent quantum images and can be extended for better compression methods on quantum images. The three types of processing operations point out patterns for designing and applying other operations on quantum images.

2 Flexible Representation of Quantum Images and Its Polynomial Preparation

Inspired by the pixel representation for images in conventional computers, a representation for images on quantum computers capturing information about colors and the corresponding positions, the quantum flexible representation of images (FRQI) is proposed. This proposal integrates information about an image into a quantum state having its formula in (1)

$$|I(\theta)\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} |c_i\rangle \otimes |i\rangle, \quad (1)$$

$$|c_i\rangle = \cos \theta_i |0\rangle + \sin \theta_i |1\rangle, \quad (2)$$

$$\theta_i \in [0, \frac{\pi}{2}], i = 0, 1, \dots, 2^{2n} - 1, \quad (3)$$

capturing information about colors and the corresponding positions of those colors, where \otimes is the tensor product notation, $|0\rangle, |1\rangle$ are 2-D computational basis quantum states, $|i\rangle, i = 0, 1, \dots, 2^{2n} - 1$ are 2^{2n} -D computational basis quantum states and $\theta = (\theta_0, \theta_1, \dots, \theta_{2^{2n}-1})$ is the vector of angles encoding colors. There are two parts in the FRQI representation of an image; c_i which encodes the information about colors and $|i\rangle$ that about the corresponding positions in the image, respectively. The quantum circuits to encode the information in an FRQI image is shown in Fig. 1. An example of a 2×2 image is shown in Fig. 2. The FRQI state is a normalized state, i.e. $\| |I(\theta)\rangle \| = 1$ as given by

$$\| |I(\theta)\rangle \| = \frac{1}{2^n} \sqrt{\sum_{i=0}^{2^{2n}-1} (\cos^2 \theta_i + \sin^2 \theta_i)} = 1. \quad (4)$$

The proposed FRQI form is quite flexible because of the way the positions of colors are encoded into computational basis states. In this way, the presentation of the

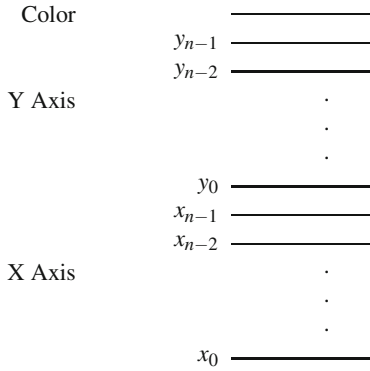


Fig. 1 The quantum circuit of FRQI representation

θ_0 00	θ_1 01
θ_2 10	θ_3 11

$$|I\rangle = \frac{1}{2} [(\cos \theta_0|0\rangle + \sin \theta_0|1\rangle) \otimes |00\rangle + (\cos \theta_1|0\rangle + \sin \theta_1|1\rangle) \otimes |01\rangle + (\cos \theta_2|0\rangle + \sin \theta_2|1\rangle) \otimes |10\rangle + (\cos \theta_3|0\rangle + \sin \theta_3|1\rangle) \otimes |11\rangle]$$

Fig. 2 A simple image and its FRQI state

geometric appearance of colors will affect on the quantum representation of the image. For example, the line by line and block based addressing methods are some of the encoding mechanisms commonly used. These mechanisms are shown in Fig. 3.

θ_0	θ_1	θ_2	θ_3
0000	0001	0010	0011
θ_4	θ_5	θ_6	θ_7
0100	0101	0110	0111
θ_8	θ_9	θ_{10}	θ_{11}
1000	1001	1010	1011
θ_{12}	θ_{13}	θ_{14}	θ_{15}
1100	1101	1110	1111

θ_0	θ_1	θ_2	θ_3
0000	0001	0100	0101
θ_4	θ_5	θ_6	θ_7
0010	0011	0110	0111
θ_8	θ_9	θ_{10}	θ_{11}
1000	1001	1100	1101
θ_{12}	θ_{13}	θ_{14}	θ_{15}
1010	1011	1110	1111

Fig. 3 Two methods for encoding position of colors

In quantum computation, computers are usually initialized in well-prepared states. As a result, the preparation process that transforms quantum computers from the initialized state to the desired quantum image state is necessary. All transforms used in quantum computation are unitary transforms described by unitary matrices. A matrix is said to be unitary if its Hermitian conjugate or its adjoint is the same as its inverse. Quantum mechanics ensure the existence of such unitary transforms for the preparation process without pointing out explicitly efficient implementation in the sense of using only simple transforms such as Hadamard transform, rotations, etc. The polynomial preparation theorem (PPT) as developed by using Lemma 1 and Corollary 1 shows a constructively efficient implementation of the preparation process. In the quantum circuit of the FRQI representation, the unitary transform \mathcal{P} to achieve the preparation process is shown in Fig. 4.

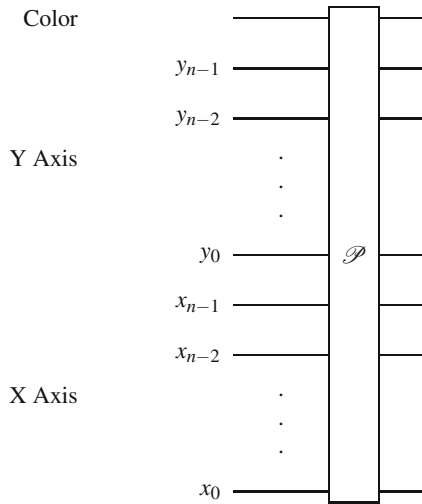


Fig. 4 The unitary transformation to achieve the preparation of FRQI images

Lemma 1. *Given a vector $\theta = (\theta_0, \theta_1, \dots, \theta_{2^{2n}-1})$ ($n \in \mathbb{N}$) of angles satisfying (3), there is a unitary transform \mathcal{P} that turns quantum computers from the initialized state, $|0\rangle^{\otimes 2n+1}$, to the FRQI state, $|I(\theta)\rangle$, composed by Hadamard and controlled rotation transforms.*

Proof. There are two steps to achieve the unitary transform \mathcal{P} as shown in Fig. 5. Hadamard transforms are used in step 1 and then controlled-rotation transforms are used in step 2 to change from $|0\rangle^{\otimes 2n+1}$ to $|H\rangle$ and then from $|H\rangle$ to $|I(\theta)\rangle$.

Let us consider the 2-D identity matrix I and the 2-D Hadamard matrix,

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

$$\begin{array}{ccc}
 |0\rangle^{\otimes 2n+1} & \xrightarrow{\mathcal{P}} & |I(\theta)\rangle = \frac{1}{2^n} \sum_{i=0}^{2^n-1} (\sin \theta_i |0\rangle + \cos \theta_i |1\rangle) \otimes |i\rangle \\
 & \searrow 1 \quad \nearrow 2 & \\
 & & |H\rangle = \frac{1}{2^n} |0\rangle \otimes \sum_{i=0}^{2^n-1} |i\rangle
 \end{array}$$

Fig. 5 Two steps to achieve the preparation operation \mathcal{P}

The tensor product of $2n$ Hadamard matrices is denoted by $H^{\otimes 2n}$. Applying the transform $\mathcal{H} = I \otimes H^{\otimes 2n}$ on $|0\rangle^{\otimes 2n+1}$ produces the state $|H\rangle$,

$$\mathcal{H}(|0\rangle^{\otimes 2n+1}) = \frac{1}{2^n} |0\rangle \otimes \sum_{i=0}^{2^n-1} |i\rangle = |H\rangle. \quad (5)$$

Let us also consider the rotation matrices (the rotations about \hat{y} axis by the angle $2\theta_i$), $R_y(2\theta_i)$, and controlled rotation matrices, R_i , with $i = 0, 1, \dots, 2^n - 1$,

$$R_y(2\theta_i) = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{pmatrix}, \quad (6)$$

$$R_i = (I \otimes \sum_{j=0, j \neq i}^{2^n-1} |j\rangle\langle j|) + R_y(2\theta_i) \otimes |i\rangle\langle i|. \quad (7)$$

The controlled rotation R_i is a unitary matrix since $R_i R_i^\dagger = I^{\otimes 2n+1}$. Applying R_k and $R_l R_k$ on $|H\rangle$ gives us

$$\begin{aligned}
 R_k(|H\rangle) &= R_k\left(\frac{1}{2^n} |0\rangle \otimes \sum_{i=0}^{2^n-1} |i\rangle\right) \\
 &= \frac{1}{2^n} \left[I|0\rangle \otimes \left(\sum_{i=0, i \neq k}^{2^n-1} |i\rangle \right) \left(\sum_{i=0}^{2^n-1} |i\rangle \right) + R_y(\theta_k) |0\rangle \otimes |k\rangle \langle k| \left(\sum_{i=0}^{2^n-1} |i\rangle \right) \right] \\
 &= \frac{1}{2^n} \left[|0\rangle \otimes \left(\sum_{i=0, i \neq k}^{2^n-1} |i\rangle \langle i| \right) + (\cos \theta_k |0\rangle + \sin \theta_k |1\rangle) \otimes |k\rangle \right], \quad (8)
 \end{aligned}$$

$$\begin{aligned}
 R_l R_k |H\rangle &= R_l (R_k |H\rangle) \\
 &= \frac{1}{2^n} \left[|0\rangle \otimes \left(\sum_{i=0, i \neq k, l}^{2^{2n-1}} |i\rangle \langle i| \right) + (\cos \theta_k |0\rangle + \sin \theta_k |1\rangle) \otimes |k\rangle + \right. \\
 &\quad \left. + (\cos \theta_l |0\rangle + \sin \theta_l |1\rangle) \otimes |l\rangle \right].
 \end{aligned} \tag{9}$$

From (9), it is clear that

$$\mathcal{R} |H\rangle = \left(\prod_{i=0}^{2^{2n-1}} R_i \right) |H\rangle = |I(\theta)\rangle. \tag{10}$$

Therefore, the unitary transform $\mathcal{P} = \mathcal{R} \mathcal{H}$ is the transform turning quantum computers from the initialized state, $|0\rangle^{\otimes 2n+1}$, to the FRQI state, $|I(\theta)\rangle$. \square

In the quantum circuit model, a complex transform is broken down into simple gates, i.e., single qubit and controlled two qubit gates, such as NOT, Hadamard, and CNOT gates which are shown in Fig. 6 and Fig. 7.

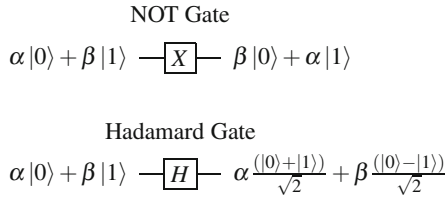


Fig. 6 NOT gate and Hadamard gate

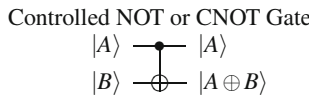


Fig. 7 CNOT gate

Corollary 1. *The unitary transform \mathcal{P} described in Lemma 1, for a given vector $\theta = (\theta_0, \theta_1, \dots, \theta_{2^{2n-1}})$, ($n \in \mathbb{N}$) of angles, can be implemented by Hadamard, CNOT and $C^{2n} \left(R_y \left(\frac{2\theta_i}{2^{2n-1}} \right) \right)$ gates, where $R_y \left(\frac{2\theta_i}{2^{2n-1}} \right)$ are the rotations about \hat{y} axis by the angle $\frac{2\theta_i}{2^{2n-1}}$, $i = 0, 1, \dots, 2^{2n} - 1$.*

Proof. From the proof of Lemma 1, the transform \mathcal{P} is composed of $\mathcal{R} \mathcal{H}$. The transforms \mathcal{H} and \mathcal{R} can be directly implemented by $2n$ Hadamard gates and 2^{2n} controlled rotations R_i or generalized- $C^{2n}(R_y(2\theta_i))$ operations, respectively. In addition, the controlled rotations R_i can be implemented by $C^{2n}(R_y(2\theta_i))$ and NOT

operations [14]. Furthermore, the result in [1] implies that $C^{2n}(R_y(2\theta_i))$ operations can be broken down into $2^{2n} - 1$ simple operations, $R_y\left(\frac{2\theta_i}{2^{2n}-1}\right)$, $R_y\left(-\frac{2\theta_i}{2^{2n}-1}\right)$, and $2^{2n} - 2$ CNOT operations. The example in the case of $n = 1$ is shown in Fig. 8.

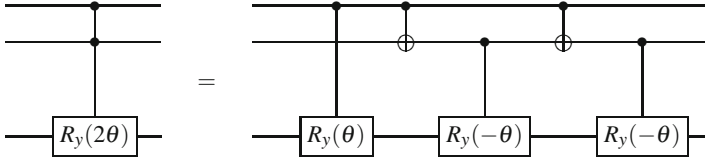


Fig. 8 $C^2(R_y(2\theta))$ gates can be built from $C(R_y(\theta))$, $C(R_y(-\theta))$ and $CNOT$ gates

The total number of simple operations used to prepare FRQI state is

$$2n + 2^{2n} \times (2^{2n-1} - 1 + 2^{2n-1} - 2) = 2^{4n} - 3 \cdot 2^{2n} + 2n. \quad (11)$$

This number is quadratic to the total 2^{2n} angle values, θ_i , $i = 0, 1, \dots, 2^{2n} - 1$. This indicates the efficiency of the preparation process. \square

Theorem 1 (Polynomial Preparation Theorem). *Given a vector $\theta = (\theta_0, \theta_1, \dots, \theta_{2^{2n}-1})$, ($n \in \mathbb{N}$) of angles, there is a unitary transform \mathcal{P} that turns quantum computers from the initialized state, $|0\rangle^{\otimes 2n+1}$ to the FRQI state,*

$$|I(\theta)\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} |c_i\rangle \otimes |i\rangle,$$

composed of polynomial number of simple gates.

Proof. Coming from Lemma 1 and Corollary 1. \square

3 Quantum Image Compression Based on Minimization of Boolean Expressions

Classical image compression techniques reduce the amount of computational resources, used to restore or reconstruct images. Similarly, quantum image compression is the procedure that reduces the quantum resources used to prepare or reconstruct quantum images. The main resource in quantum computation is the number of simple quantum gates or simple operations used in the computation. Therefore, the process, which decreases the number of simple quantum gates in the preparation and reconstruction of quantum images, is called Quantum Image Compression (QIC). The preparation and reconstruction of quantum images are the same, however, in the sense of their computation.

There are several reasons why image compression must be considered in FRQI. To start with, studies in classical image processing point out that there is redundancy in the image that can be reduced for compression in quantum image as well. Secondly, as shown in section 2, preparing a quantum image needs a large number of simple gates. For example a 2^{16} position image needs 2^{32} simple gates for preparation. Thirdly, in physical experiments, the number of simple gates should be decreased for robust implementation. For all of these reasons, the reduction of simple gates is necessary for both theoretical and practical aspects of the FRQI.

The amount of simple quantum gates used for preparing the FRQI depends mainly on the number of controlled rotation gates, as shown in Fig. 9. Essentially, most of the basic gates required for the preparation process are to simulate the $C^{2n}(\cdot)$ gates. Therefore, the reduction of controlled rotation gates results in a decrease in the total number of gates. One of the ways to reduce the number of gates is to integrate some of them under certain conditions. This section describes a method to integrate controlled rotation gates having the same rotation angle.

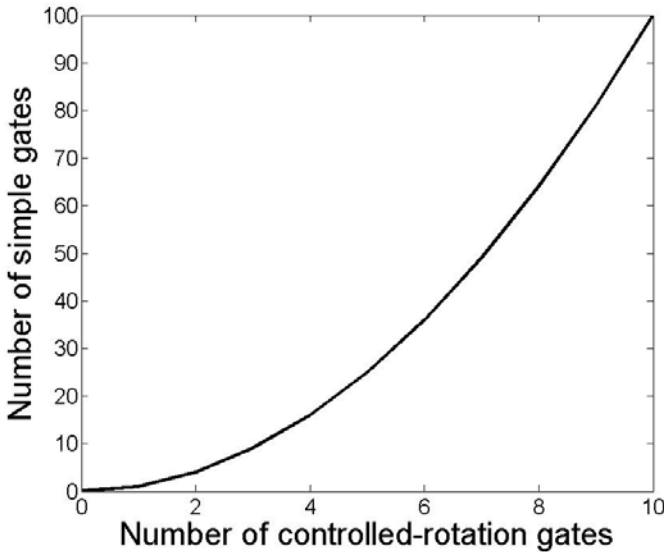


Fig. 9 The relation between number of rotation gates and total number of simple gates

As observed in classical image processing, many colors are indistinguishable to human eyes. Exploiting this fact of human vision, the classical image representations use a limited number of levels for expressing gray scales or colors in digital images with various sizes without significant impact on the quality of the images. Regarding this observation, the input angles encoding colors can take their values from a discrete set of numbers. Consequently, in the preparation process, controlled rotation operators with the same angle but different conditions affect the positions

having the same colors. Therefore, all rotations can be divided into groups such that each group includes only operators having the same rotation angle.

The difference between controlled rotation gates in one of the divided groups is only the conditional part on each gate. As presented in section 2, the conditional part of a controlled rotation gate depends on the binary string which encodes the corresponding position in an image. Therefore, the rotation angle and binary strings encoding conditional parts of rotation gates in a group characterize the group. From this point of view, each group has a generalized controlled rotation gate in which the rotation angle is the group's rotation angle and the controlled condition is the integration of all binary strings in the group.

In order to make the above arguments explicit, let us consider a 8×8 image shown in Fig. 10 as an example. This image contains only two colors, blue and red with 8 and 56 positions respectively, which requires 64 $C^6(\cdot)$ gates in general preparation discussed in section 2. Dividing all the 64 controlled rotation gates into 2 groups helps to reduce the number of gates from 64 to 4 as shown in Fig. 11 resulting in a reduction of the number of controlled-rotation gates by 93.75%. In addition, the controlled-rotation gates in the minimized circuit are much simpler than $C^6(\cdot)$ gates, implying that the number of basic gates used in each controlled-rotation gate is reduced as well. Consequently, the red-group uses one $C^1(\cdot)$ and two $C^2(\cdot)$ gates with the controlled conditions satisfy only the red positions.

There is a way to transform a binary string to a Boolean minterm by considering each position in the binary string as a Boolean variable. If x is the Boolean variable at a position in the string and the value of that position is 1 then the lateral x is used in the minterm otherwise the lateral \bar{x} is used. For example, the binary string 000 and 101 are equivalent to $\bar{x}_2\bar{x}_1\bar{x}_0$ and $x_2\bar{x}_1x_0$, respectively. With this method, there is a one-to-one correspondence between the set of all binary strings with length

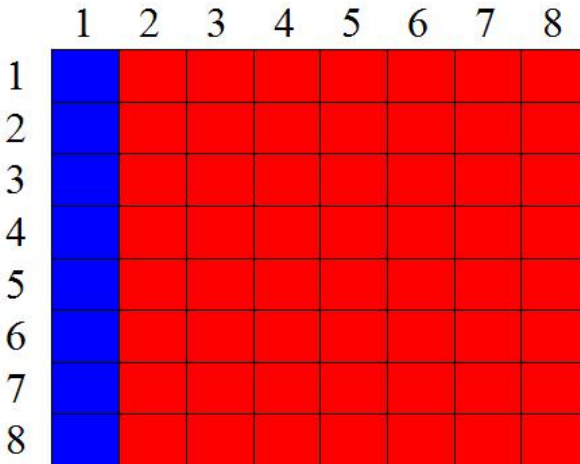


Fig. 10 8×8 two-color image

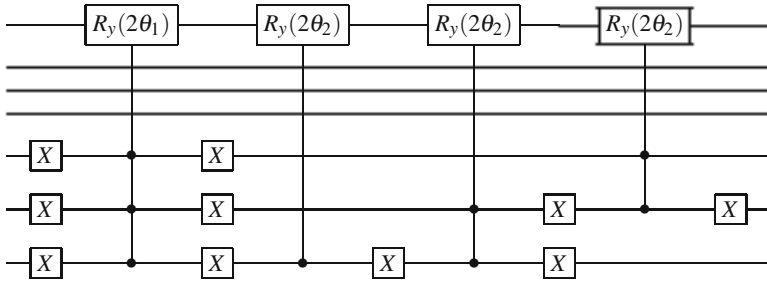


Fig. 11 The minimized circuit for 8×8 two-color image

The 8 positions having red color

$|0\rangle, |8\rangle, |16\rangle, |24\rangle, |32\rangle, |40\rangle, |48\rangle, |56\rangle$

Binary strings and Boolean minterms

$|0\rangle \rightarrow |000000\rangle \rightarrow \overline{x_5 x_4 x_3 x_2 x_1 x_0}$

$|8\rangle \rightarrow |001000\rangle \rightarrow \overline{x_5 x_4 x_3 x_2 x_1} x_0$

...

$|56\rangle \rightarrow |111000\rangle \rightarrow x_5 x_4 x_3 \overline{x_2 x_1 x_0}$

Boolean expression

$e = \overline{x_5 x_4 x_3 x_2 x_1 x_0} + \overline{x_5 x_4 x_3 x_2 x_1} x_0 + \dots + x_5 x_4 x_3 \overline{x_2 x_1 x_0}$

Minimized expression

$e = x_2 x_1 x_0$

Fig. 12 8-position group, the corresponding Boolean expression and its minimized expression

n and the set of all Boolean minterms generating from n Boolean variables. After dividing the controlled rotation gates of an image into same color groups, the next step is the compression or minimization of the gates in each group. For this purpose, the binary strings of each group play a key role. Using the method described in the above paragraph, each of these binary strings corresponds to a Boolean term. Therefore, the integration of all of binary strings in a group is equivalent to the conjunction all of their corresponding Boolean minterms. This conjunction forms a Boolean expression. For example, let us consider an 8-position group in Fig. 12 which comes from the blue-group in Fig. 10. The Boolean expression captures all information about the binary strings in the group. This means that the expression contains all information about the conditions for controlling the gates in that group. The expression can be rewritten in minimized form which contains only one term, as shown in Fig. 12. This observation suggests that only one controlled-rotation gate can be used instead of 8 gates.

The quantum image compression (QIC) algorithm is proposed to reduce controlled rotation gates in the same color groups based on the minimization of their Boolean expression as shown in Fig. 13. The procedure starts with the information

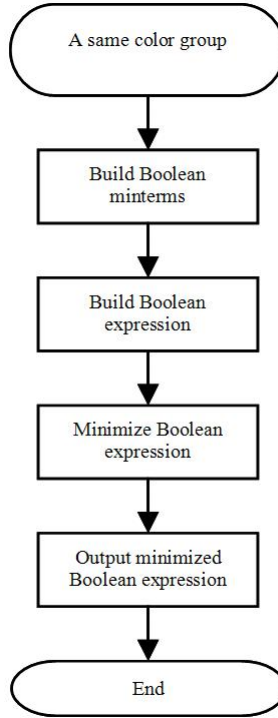


Fig. 13 Flow chart of the quantum image compression algorithm

about positions in a same color group and ends with the minimized form of the Boolean expression. The minimized Boolean expressions are used to construct a quantum circuit with a lesser number of simple gates than the original circuit. The number of product terms in a minimized Boolean expression indicates the number of conditioned rotation gates that can be used for the corresponding group of same color positions. The analysis of the compression ratio in Section 5.2 is based on this observation. The laterals in a product term in each minimized expression point out the condition part of the conditioned rotation gates. The Boolean variables with complement laterals use extra pairs of NOT gates for each complement.

In the QIC algorithm, the minimization of Boolean expressions plays a fundamental role because the number of Boolean variables in the expression is not trivial. The problem of minimizing Boolean expressions has been studied extensively, starting with Karnaugh maps, Espresso algorithm, etc [3]. The Espresso algorithm is widely used in the implementation of practical design software programs like Friday Logic, Minilog, etc. Using the Espresso algorithm, programs can minimize the Boolean expressions on 100 inputs and 100 outputs within reasonable running time. There are heuristic synthesis methods [3] that reduce the number of simple

gates in quantum circuits in general. Therefore, applying these methods after the QIC algorithm could give better results.

4 Image Processing Operators on Quantum Images Based on Invertible Transformations

Representations of images provide the background for image processing algorithms. The algorithms use an image as input to produce another image as output by performing simple operations. Furthermore the output image is analyzed to obtain useful information. This procedure in classical computers can be applied to quantum computers by using unitary transforms as image processing operations.

In classical image processing, basic operators provide fundamental manipulations in various algorithms for processing images. These operators include changes of colors at some positions, shifting the color of whole image, performing Fourier transform, etc. These basic operations are important in constructing and understanding the processing algorithms. In quantum images, however, the primary manipulations are not obvious since they should be invertible. Meanwhile, some classical operations are not invertible such as convolution operators [11] that means they are physically impossible in quantum computation. With FRQI, the basic operations can be classified and constructed by using unitary transforms.

With the FRQI proposal, images are expressed in their FRQI states and quantum image processing operations are performed using unitary transforms on those states. These transforms are divided into 3 categories; G_1 , G_2 and G_3 , applied to FRQI states dealing with only colors, colors at some specific positions and the combination of colors and positions, respectively. The first two type of operators are simple in the sense that the appearance of the output and input images is highly related. The last type is more complex because the combination of both color and position in output images make the interpretation of measurements on the output images difficult.

Operators in the first group use only information about the color, such as color shifting and the second group contains those based on colors at some position in the images, for instance the changes in color at specific positions. The last group targets information about both color and position as in Fourier transform. Each category has its own type of unitary transform. The unitary transforms are in the following forms

$$G_1 = U_1 \otimes I^{\otimes n}, \quad (12)$$

$$G_2 = U_2 \otimes C + I \otimes \bar{C}, \quad (13)$$

$$G_3 = I \otimes U_3, \quad (14)$$

where U_1, U_2 are single qubit transforms, U_3 is an n -qubit transform, C and \bar{C} are matrices regarding eligible and ineligible positions, $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ is the identity operator and n is the number of qubits encoding positions. From the point of quantum circuit modeling, G_1 uses a single qubit gate U_1 , G_2 uses an additional control from the position on the gate U_2 and G_3 just use the n -qubit transform U_3 . These circuits are shown in Fig. 12, 13, and 14, respectively.

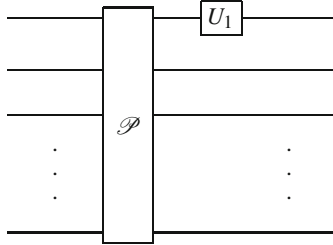


Fig. 14 Quantum circuit of G_1 operations dealing with only the color part by single qubit gates U_1

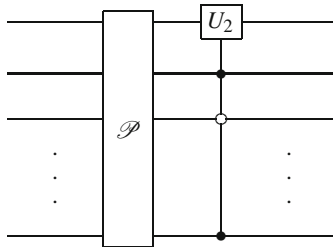


Fig. 15 Quantum circuit of G_2 operations dealing with the colors at some positions by single qubit gates U_2 .

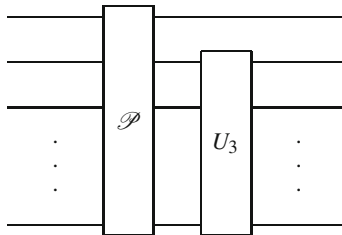


Fig. 16 Quantum circuit of G_3 operations combining both color and position information by n -qubit gates U_3 .

The color shifting operator, S , is defined as an operator in the group G_1 , $S = U \otimes I^{\otimes n}$, by using rotation matrices $U = R_y(2\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$, where θ is the

shifting parameter. The following calculation produces the result $|I_S\rangle$ of the application of S on $|I\rangle$,

$$\begin{aligned} |I_S\rangle &= S(|I\rangle) \\ &= (U \otimes I^{\otimes n}) \left(\frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} |c_i\rangle \otimes |i\rangle \right) \\ &= \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} (\cos(\theta_i + \theta)|0\rangle + \sin(\theta_i + \theta)|1\rangle) \otimes |i\rangle. \end{aligned} \quad (15)$$

The quantum image $|I_S\rangle$ has all of its colors coming from the original image $|I\rangle$ by shifting the θ angle.

The change in color at some points in an image depends on the specific positions in the image. Information about the positions is used as conditions encoded in the matrix C of the controlled-gate G_2 to construct the processing operators. For instance, let us consider a 2×2 image and the change in color at positions $|0\rangle$, $|2\rangle$. The matrix $C = |0\rangle\langle 0| + |2\rangle\langle 2|$ and $\bar{C} = |1\rangle\langle 1| + |3\rangle\langle 3|$ are used to construct the controlled-gate G_2 ,

$$G_2 = U \otimes (|0\rangle\langle 0| + |2\rangle\langle 2|) + I \otimes (|1\rangle\langle 1| + |3\rangle\langle 3|). \quad (16)$$

The action of this particular G_2 on a general 2×2 image in FRQI form, $|I\rangle = \frac{1}{2} \sum_{i=0}^3 (\cos \theta_i |0\rangle + \sin \theta_i |1\rangle) \otimes |i\rangle$, is given by

$$\begin{aligned} G_2|I\rangle &= \frac{1}{2} [(\cos \theta_0 \times U|0\rangle + \sin \theta_0 \times U|1\rangle) \otimes |0\rangle + \\ &\quad + (\cos \theta_2 \times U|0\rangle + \sin \theta_2 \times U|1\rangle) \otimes |2\rangle + \\ &\quad + (\cos \theta_1 |0\rangle + \sin \theta_1 |1\rangle) \otimes |1\rangle + \\ &\quad + (\cos \theta_3 |0\rangle + \sin \theta_3 |1\rangle) \otimes |3\rangle]. \end{aligned} \quad (17)$$

The calculation in (16) shows that the action of operator U for changing color has affects only on the specific positions $|0\rangle$, $|2\rangle$.

One of the examples on the G_3 operators, which combine both colors and positions in output images, is the operator based on quantum Fourier transform. The application of QFT on FRQI can be considered as the application of Fourier transform on the cosine part and sin part of the FRQI state coefficients as in the following calculations of c_k and s_k .

$$\begin{aligned} |T\rangle &= \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} (\cos \theta_i |0\rangle + \sin \theta_i |1\rangle) \otimes QFT(|i\rangle) \\ &= \frac{1}{2^n} \left[\sum_{k=0}^{2^{2n}-1} c_k |0k\rangle + \sum_{k=0}^{2^{2n}-1} s_k |1k\rangle \right], \end{aligned} \quad (18)$$

where

$$c_k = \frac{1}{2^n} \sum_{k=0}^{2^{2n}-1} e^{2\pi jik/2^{2n}} \cos \theta_i, \quad (19)$$

$$s_k = \frac{1}{2^n} \sum_{k=0}^{2^{2n}-1} e^{2\pi jik/2^{2n}} \sin \theta_i, \quad (20)$$

$$k = 0, 1, \dots, 2^{2n} - 1.$$

The complexity of each type of operation is specified based on the number of simple gates in the corresponding quantum circuit. The number of simple gates used for an operation in the G_1 category is one gate as in Fig. 12 that means the complexity of the G_1 operation is $O(1)$. The number of controlled rotations used for an operator in the G_2 category depends linearly on the number of positions involving the operator, $O(N)$, where N is the number of positions in the whole image. The complexity of G_3 operations depends on the complexity of the n -qubit operations U_3 as shown in Fig. 14. If we use the U_3 operations with $O(\log^2(N))$ like quantum Fourier transform, quantum Wavelet transform, etc. then the complexity of the G_3 operation is $O(\log^2(N))$.

5 Experiments on Quantum Images

A desktop computer with Intel Core 2 Duo 1.86GHz CPU and 2GB RAM is used to simulate the experiments on quantum images. The simulations are based on linear algebra with complex vectors as quantum states and unitary matrices as unitary transforms using Matlab. The simulations are based on linear algebra with complex vectors as quantum states and unitary matrices as unitary transforms using Matlab. The final step in quantum computations is the measurement which converts the quantum information into the classical form as probability distributions. Extracting and analyzing the distributions gives information for retrieving quantum images and revealing structures in these images. In section 3, the QIC algorithm reduces the number of conditioned rotation gates in quantum image storage process. The experiment on the analysis of how many gates are reduced (or compression ratio) is done using the Lena image. The minimization part in the QIC is done by Logic Friday software which is widely used in practice. The application of QFT to combine information on colors and positions on FRQI as presented in section 4 is used to detect lines in binary images.

5.1 Storage and Retrieval of Quantum Images

The essential requirements for representing a classical or quantum image are the simplicity and efficiency in the storage and retrieval of the image. The storage of a quantum image is achieved by the preparation process which is ensured by the proposed PPT in section 2. The measurement of the quantum image state produces



Fig. 17 The image used in experiments of storage and retrieval quantum images and its enlarged 8×8 lower-right block

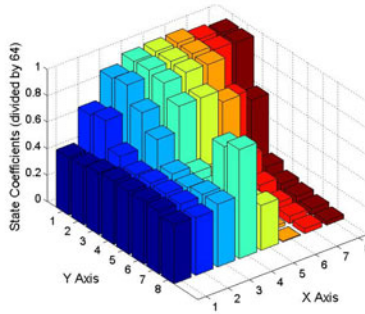


Fig. 18 64 coefficients of the 8×8 lower-right block from the input image

a probability distribution that is used for the retrieval of the image. Input information for preparation in this experiment is the gray levels coming from the following 64×64 gray image.

From the image, the angles encoding the gray levels and corresponding positions are extracted. The conditioned-rotation gates used in the quantum circuit are built based on this data. The quantum image state has 8192 complex numbers as its coefficients, Fig. 16 shows the 64 coefficients of the 8×8 lower-right block from the input image.

A measurement of a quantum state based on the set of basis vectors produces only one result which is one of the basis vectors. In quantum computation, measurements are performed on identical states instead of one state. With only one quantum state, it is impossible to get information from that state. Therefore, a measurement process

needs many identical quantum states. For instance, in order to retrieve information about the quantum state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (21)$$

many identical states, $|\psi\rangle$, are prepared. Each measurement on $|\psi\rangle$ gives either 0 or 1 as result. Many measurements, however, reveal either the result 0, with probability $|\alpha|^2$, or the result 1, with probability $|\beta|^2$. This implies that a measurement process on a quantum state gives information about the quantum state in form of a probability distribution.

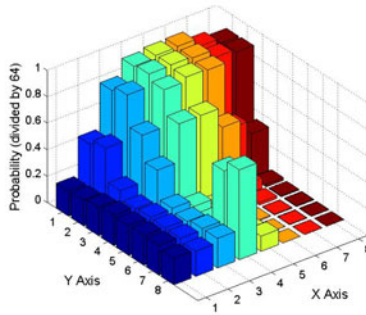


Fig. 19 Probability distribution of the 8×8 block

With general quantum states, the probability distributions are not enough to understand clearly the states because their coefficients are complex numbers. The FRQI, however, contains only real valued coefficients that make retrieval of all information about the state possible. The Fig. 17 shows the probability distribution of the 8×8 block mentioned in Fig.16. In addition, the quantum circuits indicated by the PPT provide a way to prepare many identical FRQI states used in the measurement process.

5.2 Analysis of Quantum Image Compression Ratios

As presented in section 3, the QIC method provides a way to reduce resources used in preparation and reconstruction of quantum images. In our experiments, compression ratios among groups are estimated based on the analysis of minimizations of Boolean expressions derived from the 8×4 single digit images (0-9) and the 256×256 gray image of Lena as in the Fig. 18 and Fig. 19 respectively. The minimization step in the QIC algorithm is done by Logic Friday free software. The compression ratio is as in (22).

$$\frac{\text{Rotations} - \text{reducedRotations}}{\text{Rotations}} \times 100\%, \quad (22)$$

where *Rotations* is the number of rotations indicated by the PPT and *reducedRotations* is the number of rotations indicated by the QIC algorithm.

The single digit images are binary 8×4 images of digits from 0 to 9. The quantum system for the images in the FRQI form contains 6 qubits, 5 qubits for encoding position and 1 qubit for colors. As shown in PPT, the upper bound of the number of controlled rotations to prepare the FRQI states of the images is 32 rotations. There are only 2 groups of positions having the same color because the images are binary images. The single digit images, the number of reduced rotations when QIC algorithm is applied on each group of positions and the compression ratios for each single digit image are shown in Fig. 18.

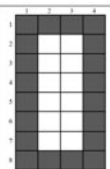
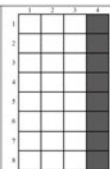
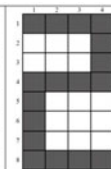
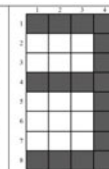
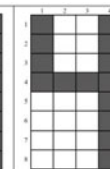
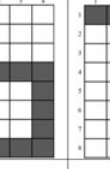


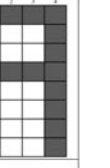

Image						
	Reduced Rotations	10	3	10	9	9
	Ratio (%)	68.75	90.63	68.75	71.86	71.86
Image						
	Reduced Rotations	10	9	8	10	10
	Ratio (%)	68.75	71.86	75.00	68.75	68.75

Fig. 20 Single digit images and their corresponding compression ratios

With experiments on the 256×256 gray scale Lena image, the quantum circuit includes 17 qubits of which 16 are used to address positions in the image and the remaining qubit is used to storing colors. The quantum circuit indicated by the PPT contains 2^{16} conditioned rotation gates. The purpose of this experiment is to analyze the compression ratio and not to deal with the preparation of the quantum state. Therefore, the preparation involving very large number of conditioned rotation gates does not matter.

A gray image can be partitioned into groups of positions having the same gray level. For the Lena image in Fig. 19, there are 207 groups containing at least one position having the same color as shown in its histogram graph in Fig. 19. The number of positions in groups ranges from one to 822 positions. The total number of conditioned rotation gates for preparation is reduced by applying the QIC for each group as shown in the upper graph in Fig. 20. The compression ratios of groups are different since the numbers and of positions and relations between the positions in

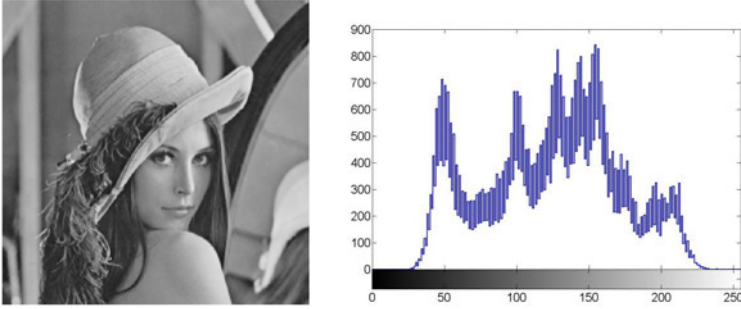


Fig. 21 Lena image and its histogram graph

Table 1 Compression ratios of same color groups in the Lena image

Positions Level	Gray Level	Rotations	Reduced Rotations	Compression Ratio (%)
50	34	77	67	12.99
100	218	105	80	23.81
150	67	151	141	6.67
200	38	207	175	15.46
250	66	253	211	16.60
700	158	702	480	31.62
750	142	775	564	27.23
800	152	806	569	29.40

groups are different. The compression ratios range from 6.67% to 31.62% between the groups as shown in Table 1 and Fig. 20.

The reasons for the variety of compression ratios between groups are

- Number of positions in the groups are different,
- The relation between positions in each group is different.

5.3 *Simple Detection of a Line in a Quantum Image Based on Quantum Fourier Transform*

Based on the discussion on QFT in section 4.2, the simulation of QFT is the application of discrete Fourier transform on a classical computer. In this experiment, each 8×8 binary image contains a line as a simple structure. These lines can be defined as periodic functions. The FRQI for the binary images used in the experiment includes 7 qubits; 6 qubits for all the positions and 1 qubit for the color. The computational basis measurements on the transformed quantum states produce the probability distribution. The detection of lines in images from the probability distributions is done by using observations on the shapes of the distributions generated

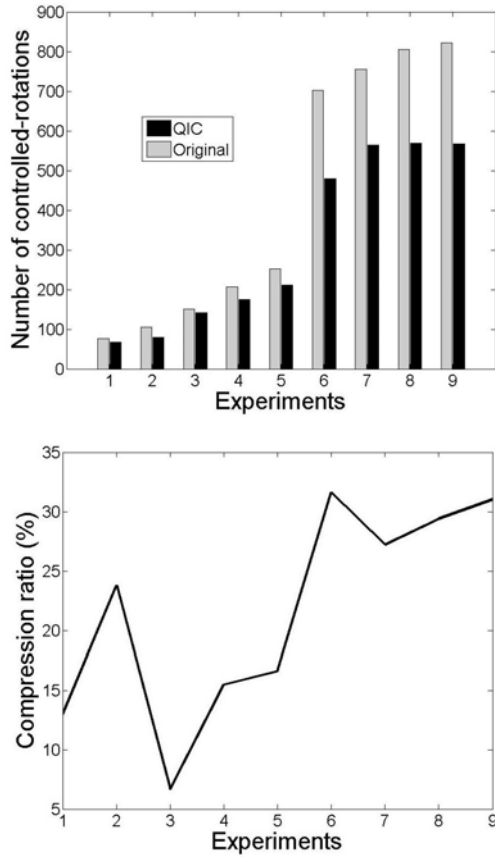


Fig. 22 The graphs of rotations and compression ratios for each group of positions having the same gray level

from cosine part of the FRQI states. The experiment studies four cases of lines in binary images shown in left side of Fig. 21.

Since all coefficients in the FRQI states are real numbers, there is a symmetric property among amplitudes of coefficients. The difference between the probability distributions in the first and fourth cases is the distance between the maximas within each distribution as shown in the right side of Fig. 21.

6 Conclusion

A flexible representation of quantum image (FRQI) is proposed in order to provide a basis for the polynomial preparation process and quantum image processing operations based on unitary operators. The FRQI captures image colors and their

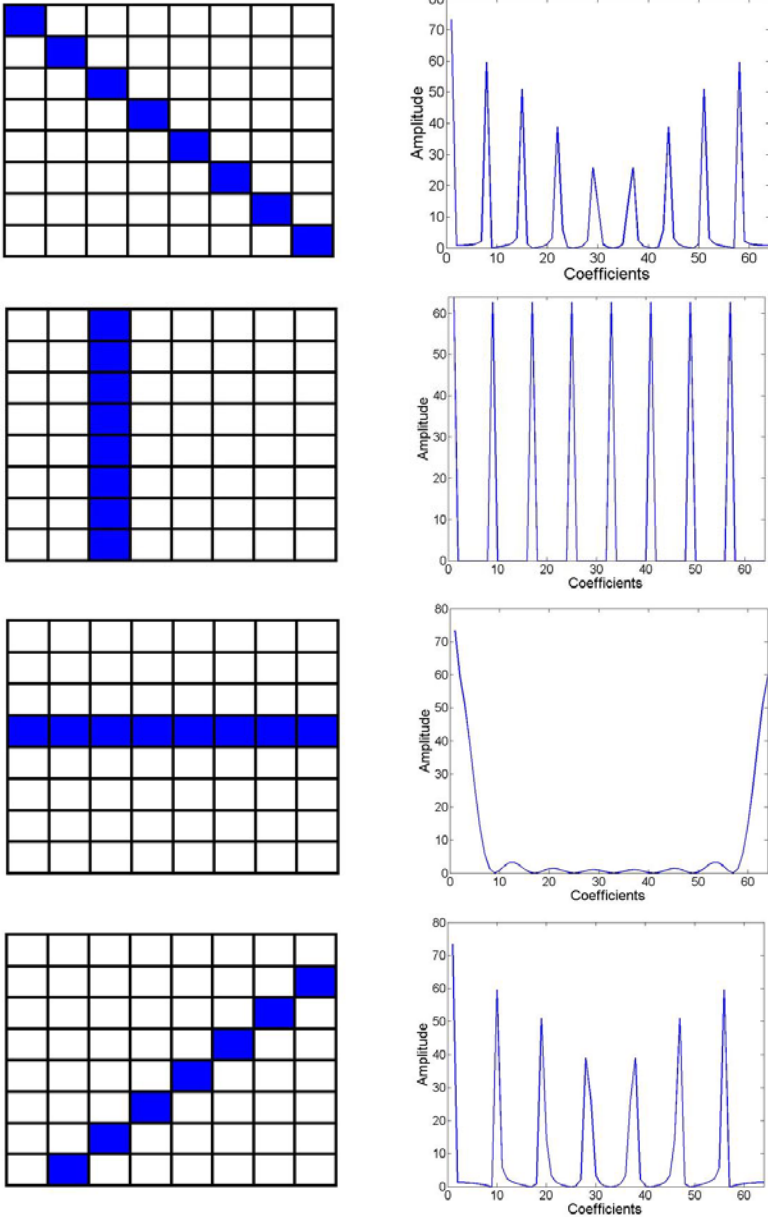


Fig. 23 8 × 8 binary images and their corresponding probability distributions

corresponding positions in a quantum state. The proposed polynomial preparation theorem (PPT) achieves a unitary preparation process using a polynomial number of simple operators transforming quantum computers from the initialized state to FRQI state. It also points out the design of the quantum circuit using Hadamard gates and controlled rotation gates for the transform. Positions in an image can be divided into groups of positions having the same color. Using the proposed quantum image compression (QIC) algorithm on the information of the groups, the number of simple gates used in FRQI preparation is reduced. The QIC is based on the minimization of Boolean equations which derives from the binary strings encoding positions in same color groups. Quantum image processing operators based on unitary transforms are addressed on FRQI. These operators are divided into 3 categories based on the 3 types of unitary transforms applied on FRQI dealing with only colors, colors at some specific positions and the combination of colors and positions. Experiments on FRQI including storing and retrieving quantum images, compression ratios of the QIC algorithm and an application of QFT as an image processing operation were done. Using the result of PPT and measurements of identical quantum states, the quantum image can be stored and retrieved. The compression ratios of the QIC algorithm on single digit binary images range from 68.75% to 90.63% and on groups having the same gray level in the Lena image range from 6.67% to 31.62%. The application of QFT in FRQI as in the detection of a line in a binary image was also discussed.

The above results imply that the FRQI can play a fundamental role in representing and processing images on quantum computers. The polynomial preparation and QIC express the efficiency of FRQI in both theory and practice. The division of three types of image processing operators on FRQI provides a guide to designing new unitary operators.

As for future work, the results presented here will be extended towards the following directions. Firstly, as the discussion in section 4, the image processing operators are divided into 3 types. The investigation on each of three types of image processing operations explores new operators on quantum images. For example, the quantum Wavelet transform, the quantum discrete cosine transform, etc. are able to replace quantum Fourier transform in the type-3 operation. Secondly, the systematic analysis of the compression ratios of QIC algorithm, presented in section 3, on a larger database of images will provide more insights into the efficiency of the algorithm. Thirdly, information-theoretic aspects on FRQI considering the existence of errors such as error-correcting codes are required for a robust representation for practical applications. The above mentioned directions are all on a single image. There are interesting questions on quantum operations having impacts on multiple images such as image matching, image searching on a set of images in FRQI states. These directions will open new results on quantum image processing in general.

References

1. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H.: Elementary gates for quantum computation. *Phys. Rev. A* 52, 3457 (1995)
2. Beach, G., Lomont, C., Cohen, C.: Quantum image processing (quip). In: *Proc. of Applied Imagery Pattern Recognition Workshop*, pp. 39–44 (2003)
3. Brayton, R.K., Sangiovanni-Vincentelli, A., McMullen, C., Hachtel, G.: *Logic minimization algorithms for VLSI synthesis*. Kluwer Academic Publishers, Dordrecht (1984)
4. Caraiman, S., Manta, V.I.: New applications of quantum algorithms to computer graphics: the quantum random sample consensus algorithm. In: *Proc. of the 6th ACM Conference on Computing Frontier*, pp. 81–88 (2009)
5. Curtis, D., Meyer, D.A.: Towards quantum template matching. In: *Proc. of the SPIE*, vol. 5161, pp. 134–141 (2004)
6. Feynman, R.P.: Simulating physics with computers. *International Journal of Theoretical Physics* 21(6/7), 467–488 (1982)
7. Fijany, A., Williams, C. P.: Quantum wavelet transform: fast algorithm and complete circuits. [arXiv:quant-ph/9809004](https://arxiv.org/abs/quant-ph/9809004) (1998)
8. Grover, L.: A fast quantum mechanical algorithm for database search. In: *Proc. of the 28th Ann. ACM Symp. on the Theory of Computing (STOC 1996)*, pp. 212–219 (1996)
9. Klappenecker, A., Rötteler, M.: Discrete cosine transforms on quantum computers. In: *Proc. of the 2nd Inter. Symp. on Image and Signal Processing and Analysis*, pp. 464–468 (2001)
10. Latorre, J. I.: Image compression and entanglement. [arXiv:quant-ph/0510031](https://arxiv.org/abs/quant-ph/0510031) (2005)
11. Lomont, C.: Quantum convolution and quantum correlation algorithms are physically impossible. [arXiv:quant-ph/0309070](https://arxiv.org/abs/quant-ph/0309070) (2003)
12. Lomont, C.: Quantum circuit identities. [arXiv:quant-ph/0307111](https://arxiv.org/abs/quant-ph/0307111) (2003)
13. Maslov, D., Dueck, G.W., Miller, D.M., Camille, N.: Quantum circuit simplification and level compaction. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 27(3), 436–444 (2008)
14. Nielsen, M., Chuang, I.: *Quantum computation and quantum information*. Cambridge University Press, New York (2000)
15. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: *Proc. 35th Ann. Symp. Foundations of Computer Science*, pp. 124–134. IEEE Computer Soc. Press, Los Almitos (1994)
16. Tseng, C.C., Hwang, T.M.: Quantum circuit design of 8×8 discrete cosine transforms using its fast computation flow graph. In: *ISCAS 2005*, vol. I, pp. 828–831 (2005)
17. Venegas-Andraca, S. E., Ball, J. L.: Storing Images in entangled quantum systems. [arXiv:quant-ph/0402085](https://arxiv.org/abs/quant-ph/0402085) (2003)
18. Venegas-Andraca, S.E., Bose, S.: Storing, processing and retrieving an image using quantum mechanics. In: *Proc. of the SPIE Conf. Quantum Information and Computation*, pp. 137–147 (2003)

Weakly Supervised Learning: Application to Fish School Recognition

Riwal Lefort, Ronan Fablet, and Jean-Marc Boucher

Abstract. This chapter deals with object recognition in images involving a weakly supervised classification model. In weakly supervised learning, the label information of the training dataset is provided as a prior knowledge for each class. This prior knowledge is coming from a global proportion annotation of images. In this chapter, we compare three opposed classification models in a weakly supervised classification issue: a generative model, a discriminative model and a model based on random forests. Models are first introduced and discussed, and an application to fisheries acoustics is presented. Experiments show that random forests outperform discriminative and generative models in supervised learning but random forests are not robust to high complexity class proportions. Finally, a compromise is achieved by taking a combination of classifiers that keeps the accuracy of random forests and exploits the robustness of discriminative models.

1 Introduction

Recent signal processing applications involve new problematics in machine learning. For instance, in addition to supervised learning scheme and unsupervised clustering, semi-supervised classification show the improvement brought by considering a training dataset formed by labelled and unlabelled data [4]. Semi-supervised classification is then considered when labelled data are lacking. One can consider a more general situation: the weakly supervised learning. In weakly supervised learning, the

Riwal Lefort

Ifremer, Technopol Brest-Iroise, 20280 Plouzane, France
e-mail: riwal.lefort@telecom-bretagne.eu

Ronan Fablet · Jean-Marc Boucher

Institut Telecom/Telecom Bretagne, Technopol Brest-Iroise - CS 83818,
29200 Brest Cedex, France
e-mail: ronan.fablet@telecom-bretagne.eu,
jm.boucher@telecom-bretagne.eu

label information of training data is composed of the prior for each class grouped together in a vector. The supervised learning and the semi-supervised learning are particular cases of weakly supervised learning. For instance, in supervised learning, prior vector gives 1 if the instance belongs to the considered class and 0 if not. In a same way, in semi-supervised learning, if the class is unknown the prior is equal for each class, and if the class is known it leads to a binary vector indicating 1 for the corresponding class as in supervised classification.

The field of fisheries acoustics provides weakly supervised learning schemes [22] [19] [16]. In fisheries acoustics, people try to recognize fish schools in images, the objective being to assess fish stock biomass, to study the marine ecosystem, or to carry out selective trawl catches. For example, when assessing the fish stock biomass in a given area, the oceanographic vessel covers the area to bring back species information. In figure 1-left, an area to be assessed is shown. The vessel transversal motion is schematically represented. Through the transversal motion, the vessel acquires images of the water column thanks to an acoustic sounder mounted on the hull. An example of acquired images is shown in figure 1-right. By successive vertical acoustic pulses, an echogram can be built in which acoustic echo samples are represented. The image then shows the acoustic response of each sample of the underwater space. Each sample of one fish school has different acoustic response compared to the seabed, the water, or the plankton. In the example of figure 1-right, the sea surface is visible as well as the bottom sea and some fish schools. The objective being to conceive classification models, a labelled training dataset is needed. In that sense, trawl catches are carried out to give the proportion of species in the related image. This proportion gives a prior knowledge for each fish schools of the images. As shown in figure 1-left, several trawl catches are realized during the acoustic campaign (trawl catches are represented with black points). Note that trawl catches often provide multi-class catch as a class proportion (classes being species). These species proportion sampling allows to build a training dataset of prior labelled fish schools. Once classification models are built, species biomasses are evaluated in non-labelled images thanks to a physic relation that links the backscattered acoustic energy to the biomass species. Several other examples of weakly supervised learning can be found in the field of computer vision. For instance, in computer vision people try to recognize objects in images for detecting their localization, their rotations and/or their scale [10] [24] [6] [5] [29]. The training dataset is then composed of images that contain objects and that are labelled with the indication of the presence or the absence of class in each image. Proposed models can then be based on Expectation-Maximization (EM) algorithm [28] [26] [20], on discriminative models [25] [27], or on Gaussian Markov random field [14].

In this chapter, three classification models are compared and studied. The first one is a generative model based on the EM algorithm [26] [9], the second one is a Fisher-based discriminative model that is extended to the non linear case [9], and the last one is a soft random forest [2] [17] that has been extended to weakly supervised learning. Classification models are useful in different situations. For instance, one model may provide strong accuracy but may not be robust to complex weakly supervised dataset. A procedure is then presented to combine the probabilistic classifiers

to improve classification performances. The three models are evaluated on a dataset composed of real fish schools. Experiments are carried out to evaluate both the robustness of the classification models as regards to the complexity of the training labels and the accuracy of the correct classification rate reached that is reached.

Section 2 is dedicated to notations and to the general framework. In the next sections 3, 4 and 5, the generative model, the discriminative model and the soft random forests are respectively presented. In section 6, the method that combines several classification models and improves classification performance is presented. Experiments are done in section 7 and concluding remarks close the chapter in section 8.

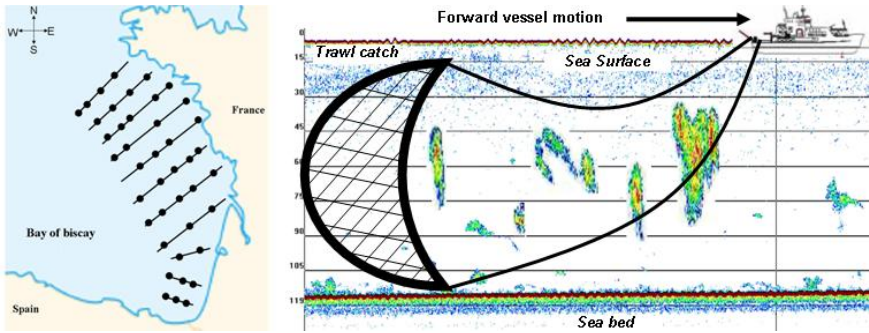


Fig. 1 In order to assess fish stock in an area (left), the vessel acquires images of the water column throughout transversal motion (left). Images contain fish schools (right) that must be classified according to their class species. Species are discriminant as a function of the shape, the position in the water column or the energy. The ground truth allowing training classification models is achieved by successive trawls catches (fishing with a net). Trawl catches spots are shown on the left with dark points.

2 Notations and General Framework

The training data is composed of objects characterized by feature vectors along with class prior vectors such that the training dataset can be written as $\{x_n, \pi_n\}_{1 \leq n \leq N}$, where $x_n = \{x_n^d\}_{1 \leq d \leq D}$ is the n^{th} object of the dataset, d being a feature index, and $\pi_n = \{\pi_{ni}\}_{1 \leq i \leq I}$ is the vector of the prior of each class i for object x_n .

We aim at defining probabilistic classification models with parameters Θ . The classification step involves the computation of the posterior $p(y = i | x, \Theta)$ for any non-labelled object x , where $y = i$ refers to the class of the object x . The classification rule typically resorts to selecting the maximum according to the posterior likelihood. Three main categories of models can be investigated:

- Generative models based on the distribution of the feature vectors for each class $p(x|y = i, \Theta)$. The required posterior probabilities are then obtained using Bayes' theorem:

$$p(y = i|x, \Theta) = \frac{p(y = i)p(x|y = i, \Theta)}{\sum_{j=1}^I p(y = j|x, \Theta)} \quad (1)$$

- The discriminative model that aims at determining hyperplans that separate classes in the descriptor space. The training consists in determining each coefficients $\Theta = \{\omega_i, b_i\}_i$ of the hyperplane that separates class i from the others, such as the posterior is given by:

$$p(y = i|x, \Theta) = \frac{\exp[\langle \Phi(x), \omega_i \rangle + b_i]}{\sum_{j=1}^I \exp[\langle \Phi(x), \omega_j \rangle + b_j]} \quad (2)$$

where $\Phi(x)$ is a function that allows to map the feature space in order to take in account non linear solutions and \langle, \rangle is the dot product.

- The soft random forests from the *boosting* family. It consists in determining a set of weak classifiers that are mixed using a vote. In this paper, the weak classifiers are soft decision trees that take probabilities at the input and provide probabilities at the output. Considering $\Theta = \{\Theta_t\}_{1 \leq t \leq T}$ where Θ_t are parameters of the t^{th} decision tree of the forest, and considering a forest that contains T decision trees, the required posterior probabilities are then obtained using the following normalizing expression:

$$p(y = i|x, \Theta) = \frac{1}{T} \sum_{t=1}^T p(y = i|x, \Theta_t) \quad (3)$$

The three approaches are detailed in the next sections.

3 Generative Model

Given $\Theta = \{\rho_{i1} \dots \rho_{iM}, \mu_{i1} \dots \mu_{iM}, \sigma_{i1}^2 \dots \sigma_{iM}^2\}$ the parameters of a Gaussian mixture model, the distribution of the feature vector for each class i is given by:

$$p(x|y = i, \Theta) = \sum_{m=1}^M \rho_{im} \mathcal{N}(x|\mu_{im}, \sigma_{im}^2) \quad (4)$$

$\mathcal{N}(x|\mu_{im}, \sigma_{im}^2)$ is the normal distribution with mean μ_{im} and a diagonal covariance matrix with component σ_{im}^2 on the diagonal. The weakly supervised learning of model parameters Θ is then stated as a probabilistic inference issue. For prior training data set of the form $\{x_n, \pi_n\}_n$ such as $\pi_{ni} = p(y_n = i)$, a maximum likelihood criterion can be derived:

$$\tilde{\Theta} = \arg \max_{\Theta} \prod_n p(\pi_n|x_n, \Theta) \quad (5)$$

We detail in this paper the solution to (5). The EM (Expectation-Maximization) procedure is exploited to estimate model parameters Θ [7]. It relies on the iterated maximization of the conditional expectation log likelihood:

$$Q(\Theta, \Theta^c) = E_y \left[\ln p(x, y | \pi, \Theta) \middle| x, \pi, \Theta^c \right] \quad (6)$$

c refers to current parameters. Assuming that objects in any image are independent, (6) can be turned into :

$$Q(\Theta, \Theta^c) = \sum_{n=1}^N \left\{ \sum_i^I p(y_n = i | x_n, \Theta^c) \ln \left[\pi_{ni} p(x_n | y_n = i, \Theta) \right] \right\} \quad (7)$$

When considering proportion-based training data, the proportion data is regarded as a class prior for each image, such that the E-step is modified to take into account this prior knowledge as follows:

$$p(y_n = i | x_n, \Theta^c) = \frac{\pi_{ni} p(x_n | y_n = i, \Theta^c)}{\sum_j \pi_{nj} p(x_n | y_n = j, \Theta^c)} \quad (8)$$

In the M-step, log-likelihood (7) is maximized with the respect to the variable Θ . Reminding that the dependency of (7) upon Θ^c is only due to $p(y_n = i | x_n, \Theta^c)$ and independently separating the maximization for each class i , the M-step amounts to maximizing a typical log likelihood weighted by $p(y_n = i | x_n, \Theta^c)$ of the Gaussian mixture model defined by (4):

$$Q_i(\Theta, \Theta^c) = \sum_{n=1}^N p(y_n = i | x_n, \Theta^c) \ln [p(y_n = i | x, \Theta)] \quad (9)$$

The maximization of (9) with respect to Θ is then issued from a second EM procedure. Introducing the hidden variable s_{ni} , defined as $p(s_{ni} = m) = \rho_{im}$ that indicates the probability for the item to be classified among the m^{th} mode of the distribution of the class i , the conditional expectation log likelihood is maximized:

$$Q_i^*(\theta, \theta^c) = E_s \left[\ln \left(p(x, s | \theta) \right) \middle| x, \pi, \theta^c \right] \quad (10)$$

Where $\theta = \{\mu_{i1} \dots \mu_{iM}, \sigma_{i1} \dots \sigma_{iM}\}$, i.e. the mean and the variance for each mode of the Gaussian mixture for class i . Similarly to (7), the complete log likelihood (10) can be rewritten as:

$$Q_i^*(\theta, \theta^c) = \sum_{n=1}^N \left\{ p(y_n = i | x_n, \Theta^c) \sum_{m=1}^M p(s_{ni} = m | x_n, \theta^c) \ln \left[\rho_{im} \mathcal{N}(x_n | y_n = i, \theta) \right] \right\} \quad (11)$$

The E-step of the second EM algorithms is given by:

$$p(s_{ni} = m | x_n, \theta^c) = \frac{\rho_{im} \mathcal{N}(x_n | s_{ni} = m, \theta^c)}{\sum_{l=1}^M \rho_{il} p(x_n | s_{ni} = l, \theta^c)} \quad (12)$$

New parameters θ are given in the M-step, by optimization of the complete log likelihood (11) with the respect to θ . A typical Lagrange multipliers procedure is then used to compute $\{\rho_{im}\}$.

The whole algorithm is shown in table 1. In comparison to the algorithm proposed in [26] for which the presence or the absence of classes are known in training images, here the class priors π_n must not be assessed in the 3rd step of the procedure. Secondly, in comparison to the common EM procedure that considers a single hidden variable indicating the considered mode, the weakly supervised learning needs to take into account two hidden variables: y_n and s_{ni} such as $s_{ni} = m$ indicates that object x_n is classified in mode m of the multi modal distribution of class i . This constraint leads to develop two EM procedures that are mixed. This is shown in table 1 where there are two E-steps in items 1 and 2, and one M-step in item 3.

The advantages of the generative model are the solid mathematical developments and the large quantity of papers that deals with the EM procedures. Furthermore, generative models are close to data and describe the data distribution with accuracy. Drawbacks of the model are the possibility for the optimization to be in a local maximum point. Generative models are known to do not fit well in presence of noisy datasets that produce weak classification accuracy. For lots of datasets, in supervised learning, these models are then outperformed by other classification models such as Support Vector Machine (SVM) or random forest.

Table 1 Learning of the generative classification model

Given an initialization for $\Theta = \{\rho_{im}, \mu_{im}, \sigma_{im}^2\}_{i,m}$, do until convergence:

1. Update the posterior likelihood of the 1st hidden variable likelihood:

$$\tau_{ni} = p(y_n = i | x_n, \Theta) = \frac{\pi_{ni} p(x_n | y_n = i, \Theta)}{\sum_{j=1}^I \pi_{nj} p(x_n | y_n = j, \Theta)}$$

2. Update the posterior likelihood of the 2nd hidden variable likelihood:

$$\gamma_{nim} = p(s_{ni} = m | x_n, \Theta) = \frac{\rho_{im} \mathcal{N}(x_n | s_{ni} = m, \Theta)}{\sum_{l=1}^M \rho_{il} p(x_n | s_{ni} = l, \Theta)}$$

3. Update the parameters $\Theta = \{\rho_{im}, \mu_{im}, \sigma_{im}^2\}$:

$$\rho_{im} = \frac{\sum_n \tau_{ni} \gamma_{nim}}{\sum_n \tau_{ni}}, \quad \mu_{im} = \frac{\sum_n \tau_{ni} \gamma_{nim} x_n}{\sum_n \tau_{ni} \gamma_{nim}}, \quad \text{and} \quad \sigma_{im}^2 = \frac{\sum_n \tau_{ni} \gamma_{nim} (x_n - \mu_{im})(x_n - \mu_{im})^T}{\sum_n \tau_{ni} \gamma_{nim}}$$

4 Discriminative Model

4.1 Linear Model

Discriminative models are stated as an explicit parameterization of the classification likelihood. They are here defined as probabilistic versions of discriminative models. As proposed by [26] [9] [16], probabilistic linear discriminative models can be defined as follows:

$$p(y = i|x, \Theta) \propto F(\langle \omega_i, x \rangle + b_i) \quad (13)$$

where $\langle \omega_i, x \rangle + b_i$ is the distance to the separation hyperplane defined by $\langle \omega_i, x \rangle + b_i = 0$ in the feature space. Model parameter Θ is given by $\{\omega_i, b_i\}_i$. F is an increasing function, typically an exponential or a continuous stepwise function. Hereafter, F will be chosen to be the exponential function:

$$p(y = i|x, \Theta) = \frac{\exp(\langle \omega_i, x \rangle + b_i)}{\sum_{j=1}^I \exp(\langle \omega_j, x \rangle + b_j)} \quad (14)$$

In [26], a maximum likelihood (ML) criterion is derived for the estimation of the model parameters for the presence/absence training data. The resulting gradient-based optimization was proven experimentally weakly robust to the initialization. A two-stage optimization was then developed. It exploits a Fisher-based criterion to estimate a normalized vector defining each discrimination plane. In a second step, a gradient-based optimization of the norm of this vector w.r.t. a ML criterion is carried out.

The Fisher-based discrimination is derived as follows. A "one-versus-all" strategy is considered, so we hereafter consider a two-class case. Fisher discrimination [12] amounts to maximizing the ratio between inter-class and intra-class variances:

$$\hat{\omega}_i = \arg \max_{\omega_i} \left\{ \frac{\omega_i^T (m_{i1} - m_{i2})}{\omega_i^T (\Sigma_{i1} + \Sigma_{i2}) \omega_i} \right\} \quad (15)$$

where m_{i1} and Σ_{i1} are the mean and variance of the class i , and m_{i2} and Σ_{i2} are the mean and variance of the remaining classes. The estimate is given by $\hat{\omega} = (\Sigma_{i1} + \Sigma_{i2})^{-1} (m_{i1} - m_{i2})$.

Fisher discrimination is applied to weakly supervised learning based on the estimation of class mean and variance for known object class priors. Formally, for a given class i , mean m_1 is estimated as:

$$m_{i1} \propto \sum_n^N \pi_{ni} x_n \quad (16)$$

m_{i2} are computed replacing π_k by $(1 - \pi_k)$, Σ_{i1} and Σ_{i2} are calculated identically:

$$\Sigma_{i1} \propto \sum_n^N \pi_{ni} (x_n - m_1)(x_n - m_1)^T \quad (17)$$

Once the initialization is done, in order to find the better coefficients $\tilde{\Theta}$, a minimum error criterion using a typical gradient minimization is considered:

$$\tilde{\Theta} = \arg \min_{\Theta} \sum_k D(\tilde{\pi}_k(\Theta), \pi_k) \quad (18)$$

where $\tilde{\pi}_k(\Theta)$ and π_k are respectively the vector of the estimated class priors in image k and the real class priors in image k , and D a distance between the observed and estimated priors. Among the different distances between likelihood functions, the Battacharrya distance [1] is chosen:

$$D(\tilde{\pi}_k(\Theta), \pi_k) = \frac{1}{N} \sum_{k=1}^N \sqrt{\tilde{\pi}_k(\Theta) \cdot \pi_k} \quad (19)$$

The major drawback of this basic model is that the non linear separations of classes are not taken into account.

4.2 Non Linear Model

A non-linear extension of the model defined by (13) can be derived using a kernel approach. The non linear mapping using kernel trick [23] [9] is based on the Kernel principal component analysis method (Kpca). It consists in a transformation of the feature space in which linear solutions are difficult to obtain. In the mapped space, a linear model is specified. The expression of the posterior is then as follows:

$$p(y = i|x, \Theta) \propto F(\langle \omega_i, \Phi(x) \rangle + b_i) \quad (20)$$

The "kernel trick" is that the function $\Phi(x)$ must not be known explicitly, but only the dot product $\langle \Phi(x1), \Phi(x2) \rangle$ defined by kernel function $K(x1, x2) = \langle \Phi(x1), \Phi(x2) \rangle$. Here, a Gaussian kernel with scale parameter a is chosen:

$$\langle \Phi(x1), \Phi(x2) \rangle = \exp\left(\frac{-\|x1 - x2\|^2}{2a^2}\right) \quad (21)$$

In order to reduce the space dimensionality, the kernel trick is associated to a principal component analysis (PCA) whose size is N_{pca} (see table 2). This model is very similar to the SVM. In comparison to SVM that maximizes merges in the mapped space [23], the weighted Fisher criterion is here used in the mapped space. The whole procedure including the non linear mapping and the parameters assessment is given in table 2.

Table 2 Learning of the non-linear discriminative classification model

Given a training dataset $\{x_n, \pi_n\}_{1 \leq n \leq N}$, do:

1. Computation of the covariance matrix:

$$K = \{K(x_n, x_m)\} = \exp\left(\frac{-\|x_n - x_m\|^2}{2\sigma^2}\right)$$

2. Diagonalization of the covariance matrix:

$$N\lambda\alpha = K\alpha$$

where $\lambda = \{\lambda^d\}_d$ are eigen values (sorted by order) and $\alpha = \{\alpha^d\}$ are eigen vectors.

3. Projection of training instances in the mapped space:

$$\Phi(x_n)^d = \sum_{m=1}^{Npca} \alpha_m^d K(x_m, x_n)$$

where d denotes the feature index in the mapped space, $Npca$ denotes the size of the truncated mapped space, and α_m^d denotes the components of the d^{th} eigen vector of the covariance matrix K .

4. Computation of the linear separation hyperplans in the mapped space for each class i :

$$\omega_i = (\Sigma_{i1} + \Sigma_{i2})^{-1}(m_{i1} - m_{i2}) \text{ and } b_i = \omega_i(\Sigma_{i1} + \Sigma_{i2})/2.$$

5. Optimization of the linear separation hyperplans in the mapped space for each class i :

$$\tilde{\Theta} = \arg \min_{\Theta} \sum_k D(\tilde{\pi}_k(\Theta), \pi_k).$$

The advantages of the discriminative model are the good performance reached, the robustness of the parameterized posterior function and the flexibility in use regarding to the kernel choice and associated parameters. The drawbacks are the same than the SVM, i.e. a possibility for the optimization to find a local minimum point, the kernel choice that can not be matched to the considered dataset, and the difficulty to interpret the data, especially in the mapped space.

5 Soft Decision Trees and Soft Random Forests

5.1 Soft Decision Trees

Decision trees are classification models that sample the feature space in homogeneous groups. This unstable classifier is well used with random forests that generate several trees and reduce the instability.

Learning a classification tree involves an iterative procedure which sequentially creates children nodes from the terminal nodes of the current iteration. At each node, the corresponding cluster of objects is splitted in several homogeneous groups. This procedure is typically carried out until children groups reach some predefined level of class homogeneity. Known methods propose different criterions to split instances in homogeneous groups [3] [21] [15] [18].

Formally, at a given parent node, the attribute and associated split value are determined with respect to the maximization of some information gain G :

$$\arg \max_{\{d, S_d\}} G(S_d) \quad (22)$$

where d indexes attributes and S_d is the split value associated to the attribute d . The Shannon entropy of object classes is among the popular gain criterion [21]:

$$\begin{cases} G = \left(\sum_m E^m \right) - E^0 \\ E^m = - \sum_i p_{mi} \log(p_{mi}) \end{cases} \quad (23)$$

where E^0 indicates the entropy at the parent considered node, E^m is the entropy obtained at the children node m , and p_{mi} the likelihood of the class i at node m . Regarding the classification step, an unlabelled object passes through the decision tree and is assigned to the class of the terminal node that it reaches.

We here present a criterion to build classification trees in a weakly supervised context. From the original C4.5 scheme [21], an entropy-based splitting criterion computed from class priors instead of class labels is proposed. It relies on the evaluation of likelihoods p_{mi} of object classes i for children nodes m . A first solution might be to consider the mean of the class likelihoods over all the instances in the considered cluster. It should however be noted that class priors can be interpreted as classification uncertainties for each training sample. Consequently, the contributions of samples with low and high uncertainties are expected to be weighted. For instance, samples associated with a uniform prior should weakly contribute to the computation of the class priors at the cluster level. In contrast, a sample known to belong to a given class provides a particularly informative prior. For feature index d , denoting x_n^d the feature value for sample n and considering the children node m_1 that groups together data such as $\{x_n^d\}_n < S_d$, the following fusion rule is then proposed:

$$p_{m_1 i} \propto \sum_{\{n\} | \{x_n^d\} < S_d} (\pi_{ni})^\alpha \quad (24)$$

For the second children node m_2 that groups data such as $\{x_n^d\}_n > S_d$, the equivalent fusion rule is suggested:

$$p_{m_2 i} \propto \sum_{\{n\} | \{x_n^d\} > S_d} (\pi_{ni})^\alpha \quad (25)$$

The considered power exponent α weights low-uncertain samples, i.e. samples such that class priors closer to 1 should contribute more to the overall cluster mean p_{mi} . An infinite exponent values resorts to assign the class with the greatest prior over all samples in the cluster. In contrast, an exponent value close to zero withdraws low class prior from the weighted sum. In practice, we typically set α to 0.8. This setting comes to give more importance to priors close to one. If $\alpha < 1$, high class priors are given a similar greater weight compared to low class priors. If $\alpha > 1$, the closer to one the prior, the greater the weight.

Note that in comparison to previous work, final nodes are associated to prior vector instead of integer indicating the class.

The procedure to train a soft tree is given in table 3.

Table 3 Learning of the soft random forests

Given a training dataset $\{x_n, \pi_n\}_{1 \leq n \leq N}$, learn T soft decision trees as follows:

1. At a given children node m that is not identified as a final node and that is not split again, find the split value S_d and the descriptor d that maximize G :

$$G = - \sum_{i=1}^I \left[\sum_{\{n\}|\{x_n^d\} < S_d} (\pi_{ni})^\alpha \log \left(\sum_{\{n\}|\{x_n^d\} < S_d} (\pi_{ni})^\alpha \right) + \sum_{\{n\}|\{x_n^d\} > S_d} (\pi_{ni})^\alpha \log \left(\sum_{\{n\}|\{x_n^d\} > S_d} (\pi_{ni})^\alpha \right) \right]$$

2. Split the data in two groups $\{x_n | x_n^d < S_d\}$ and $\{x_n | x_n^d > S_d\}$ respectively associated to children nodes m_1 and m_2 .

3. Compute the class priors $p_{m_1} = \{p_{m_1 i}\}_i$ in children node m_1 and the class priors $p_{m_2} = \{p_{m_2 i}\}_i$ in children node m_2 such as:

$$p_{m_1 i} \propto \sum_{\{n\}|\{x_n^d\} < S_d} (\pi_{ni})^\alpha \text{ and } p_{m_2 i} \propto \sum_{\{n\}|\{x_n^d\} > S_d} (\pi_{ni})^\alpha$$

4. If the children node m_1 is class-homogeneous enough, then m_1 is a final node with associated class prior p_{m_1} .

If the children node m_2 is class-homogeneous enough, then m_2 is a final node with associated class prior p_{m_2} .

5. If there exists node m that are not final nodes return to step 1 and treat them.

5.2 Soft Random Forest

Whereas the unsteadiness of one tree is a critical issue, boosting procedures can exploit this drawback to build ensemble classifiers to reach remarkable classification performance [8] [2] [13]. The randomization of classification trees, especially

random forests [2], have been shown to be a powerful and flexible tool for improving classification performances. This randomization may occur at different levels: in the random selection of subsets of the training dataset, in the random selection of the feature space, in the random selection of the features considered for each splitting rule. The classification step generally comes to a voting procedure over all the generated trees.

Once a tree is built from weakly labelled data, a random forest [2] can be elaborated in the same way. Trees are not pruned. Let t , $1 \leq t \leq T$ be the tree index for the created random forests.

Regarding the classification of unknown samples, we proceed as follows. A test instance x goes through all the trees of the forest. As a result, the output from each tree t is a prior vector $p_t = [p_{t1} \dots p_{ti}]$. p_t is the class probability at the terminal node of the tree t . The probability that x is assigned to class i , i.e. the posterior likelihood $p(y = i|x)$, is then computed as a mean:

$$p(y = i|x) = \frac{1}{T} \sum_{t=1}^T p_{ti} \quad (26)$$

6 Classifier Combination

In this section, a combination of classifiers is investigated. Different experimental properties can be expected from the considered classifiers, especially random forest and discriminative models, in terms of robustness to the complexity of the training data. The latter models might be more robust to uncertainties, and thus to complex training mixtures, as they rely on a parametric (linear) estimation of the separation planes between object classes. In contrast, random forests potentially depict greater adaption capabilities. This property may become a drawback for datasets with larger training uncertainties. Then it should be appropriate to combine posteriors from different classifiers in order to extract positive information.

Let Θ_1 and Θ_2 be the parameters of two assessed classifiers and let $p(y = i|x, \Theta_1)$ and $p(y = i|x, \Theta_2)$ be their posterior classification likelihoods. Two approaches might be undertaken to exploit the two posteriors:

- A way may be to use the usual classifier combination that is expressed as follows [11]:

$$p(y = i|x, \Theta_1, \Theta_2) \propto \beta p(y = i|x, \Theta_1) + (1 - \beta) p(y = i|x, \Theta_2) \quad (27)$$

where β is a parameter that gives less or more weight to each classifier. For example, if Θ_1 and Θ_2 are respectively the parameters of the discriminative model and the random forests, β will set a compromise between the robustness of the discriminative model as regard to the high complexity labels and the random forests as regard to the high accuracy reached in supervised learning.

- An other way will be to update the prior with a classifier and use the updated prior to train an other classifier. Formally, we proceed as follows. Given a probabilistic classifier with parameters Θ_1 , we compute the resulting posterior classification

likelihoods $\{p(y_n = i|x_n, \Theta_1)\}_{n,i}$ for any training sample x_n . Given the training prior $\pi_n = \{\pi_{ni}\}$ for sample x_n , this prior is updated as:

$$\pi_{ni}^{new} \propto p(y_n = i|x_n, \Theta_1)\pi_{ni}^\beta \quad (28)$$

Finally, this new training prior is considered to learn the final classifier with parameters Θ_2 . The considered training dataset is then $\{x_n, \pi_n^{new}\}_n$. Coefficient β states the relative confidence in the posterior issued from the classifier Θ_1 w.r.t. the initial training prior. It might be noted that this fusion rule guarantees that impossible classes for a given sample (i.e. classes associated with a null prior) remain excluded. In particular, the prior labelled samples, i.e. priors equalling 1 for one class, will not be modified by this update. This procedure is particularly relevant for training samples with highly uncertain priors.

In the experiments the second proposed solution will be chosen with Θ_1 being the parameters of a discriminative model and Θ_2 the parameters of a soft random forests. The drawback of the first solution is that prior training knowledge, such as $p_{i_{ni}} = 0$, are not conserved.

7 Application to Fisheries Acoustics

7.1 Simulation Method

In practice, because the ground truth is only composed of the proportion of classes in images, no one can know exactly the individual class of each object in the images. Weakly supervised training dataset are then built from supervised training dataset.

The procedure to build a weakly supervised training dataset from a given supervised dataset is reported in table 4. We distribute all the training examples in several groups according to predefined target class proportions. All the instances in a given group are assigned to the class proportion of the group. In table 4, examples of target proportions are shown for a four-class dataset. The objective being to evaluate the comportment of classification models as regard to the complexity of the class mixture, we create groups containing from one class (supervised learning) to the maximum-class available (four classes in the example of table 4). For each case of class-mixture, different mixture complexities can be created: from one class dominating the mixture, i.e. the prior of one class being close to one, to equiprobable class, i.e. nearly equal values of the priors. For example, in table 4, considering three-class mixture, 24 images are built with the corresponding class proportions.

Mean classification rates are assessed using a cross validation procedure over 100 tests. 90% of data are used to train classifier while the 10% remainders are used to test. Dataset is randomly split every test and the procedure that affects weak labels to the training data is carried out at each test. For each test of the cross validation, the correct classification rate corresponds to the mean of the correct classification rate per class.

with swim bladder has a more important backscattering strength than fish without swim bladder.

In practice, fish schools are identified by experts from association between mono specific trawl catches and acoustic images acquired during the trawling operation. If trawl catches provide only one species, we suppose that fish schools in the corresponding images contain only the considered species.

In the database, four classes of species are identified: Sardina (179 fish schools), Anchovy (478 fish schools), Horse Mackerel (667 fish schools), and Blue Whiting (95 fish schools). For instance, different fish schools are represented in figure 2. Sardina schools appear dense and large with lot of backscattering strength, Anchovy schools are scattered from the seabed to the middle of the water column, and Horse Mackerel are rather situated close to the seabed with spatial organisation similar to Anchovy.

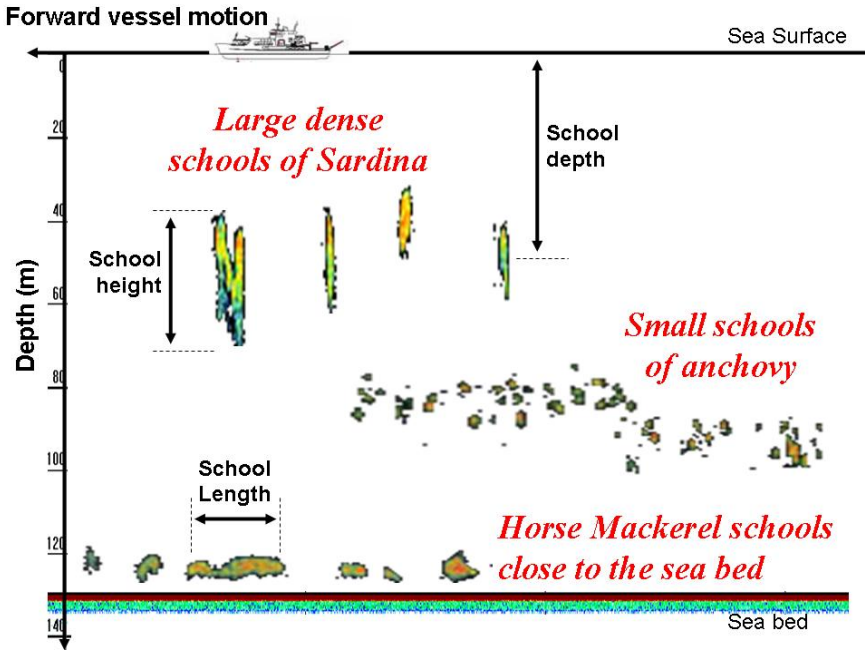


Fig. 2 Examples of fish school organisations in one echogram for Anchovy, Sardina, and Horse Mackerel

7.3 Results

Results are shown in figure 3. The mean correct classification rate is reported for the generative model (EM), for the discriminative model based only on the Fisher model (Fisher) that is presented in equation (15), for the discriminative based on

the Fisher model followed by the optimization (Fisher + Optim) that is presented in equation (18), for the soft random forest (SRF), and for the combination between SRF and Fisher (SRF + Fisher). The combination of the two classification models is carried out in applying the method proposed in section 6 with equation (28). Θ_1 are the parameters of the Fisher-based discriminative model and Θ_2 are the parameters of the random forest that is built with the dataset $\{x_n, \pi_n^{new}\}_n$. The classification rate is shown as a function of the number of class in training images from one class (supervised learning) to four classes and following the target proportion shown the table 4.

Firstly, we analyse the supervised learning to notice that, for this dataset, random forests greatly outperforms the generative and the discriminative models. Actually, the rate goes from 0.63 to 0.7 with generative and discriminative models whereas it reaches 0.9 with random forest. The high performances reached by random forest in supervised learning justified their use in a weakly supervised learning.

Secondly, looking at the weakly supervised learning, we notice that performance fall down compared to supervise learning. It is particularly true for the random forests that loose around 30% accuracy in four-class mixture compare to supervised learning and the generative model that loses around 20% accuracy in four-class mixture compared to supervised learning. For random forests the explanation is that the used criterion to find acceptable split at the corresponding node m does not fit for prior labelling. Actually, in most of cases because of mean calculation (24) and (5.1), situations may produce uniform class distribution p_m . In fact, there is no normalization term in equations (24) and (5.1) that provides information about the number of instance that are involved by each class. The falling down performances provided by the generative model can be explained by the difficulty for the EM procedure to fit with complex data. Especially when the data organisation in the descriptor space does not correspond to Gaussian mixture and when there is a lot of overlapping between classes. In comparison, the weighted Fisher-based model is more robust as regards to the prior complexity. Actually, the discriminative model is down only around 1% accuracy from the supervised learning to the four-class mixture. The simplicity of the Fisher weighting and the non linear mapping explains this robustness. The analysis of the comportment of the discriminative optimization reveals the drawback of this approach, i.e. the non-optimal convergence. A rate improvement from the weighted-Fisher was waited but there is a significant loss from 3% to 5% rate. This can be explained by the fact that a lot of solutions exist for equation (18) and there is not enough constraints to find the true solution.

On the opposite, the classifier combination seems to be a very good solution to weakly supervised data. Using equation (28) to combine the discriminative model and the random forests, high accuracy performances are reached compared to single models such as discriminative model or soft random forest. By fusing responses, the robustness of the discriminative model is kept (there is a rate loss around 2% from the supervised learning to the three class mixture and around 10% from the supervised learning to the four-class mixture) and the high accuracy reached by the random forests is conserved too (the correct classification rate goes from 89.2% in the supervised learning case to 77.2% in the four-class mixture case).

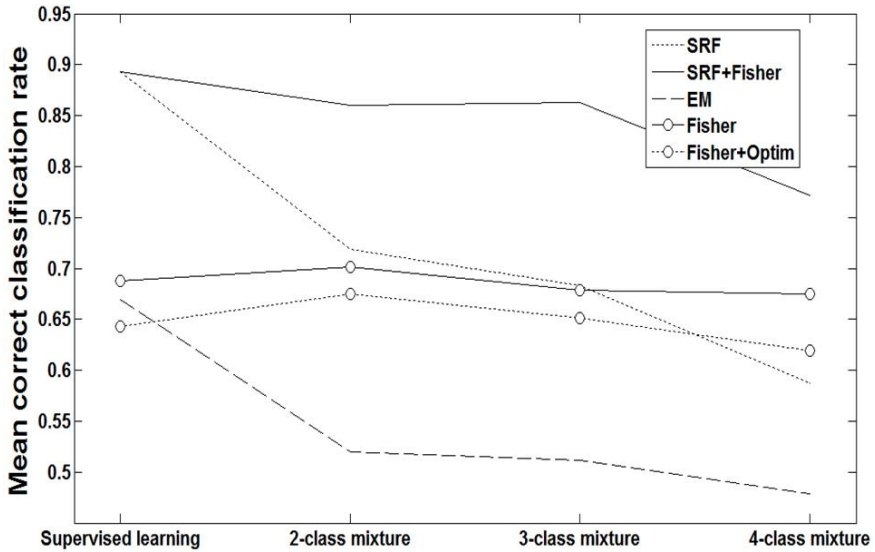


Fig. 3 Mean correct classification rate as a function of the number of class per training images

In figure 4, two confusion matrixes are shown for the classification model that combine the soft random forests with the discriminative model. The confusion matrixes are reported for the supervised learning (figure 4-left) for which the mean correct classification rate equals 0.893 and the four-class mixture (figure 4-right) for which the mean correct classification rate equals 0.772. Note that confusion matrixes are obtained by computing the mean over the cross validation which explains that horizontal and vertical sums do not exactly equal to 1. In the supervised learning case, correct classification rates per class reach high performance except for *Sardina* that provides a mean correct classification rate that equals 73.8%. Blue Whiting seems to be the class that is well separated from the others with a correct classification rate of 97%. In the four-class mixture case, the *Sardina* does not change and the correct classification rate of the other classes fall down from around 15%.

While the combination of the random forests and of the discriminative models resorts to the best performances, we further analyse the robustness of each classifier. In figure 5, we report classification performances w.r.t. mixture complexity. We evolve the complexity of the 3-class training mixture from the supervised case to the unsupervised case (i.e. uniform prior). Note that for each experiment all training samples are generated with the same type of mixture proportion (see table 4), i.e. the training data does include both low and high uncertainty samples. These results clearly illustrate the relative robustness of the different classifiers to the degree of class uncertainty in the training dataset. Obviously, classification performance decreases in all cases. The slopes are however different. Whereas the classification trees greatly outperform the two other types of classifiers in the supervised case, it also shown to

Supervised learning					Four-class mixture				
	Sardina	Anchovy	Horse Maquerel	Blue Whiting		Sardina	Anchovy	Horse Maquerel	Blue Whiting
Sardina	73.8%	12.7%	13.3%	0%	Sardina	76.6%	3.3%	2%	0%
Anchovy	0.4%	96%	3.5%	0%	Anchovy	7.7%	72.9%	19.3%	0%
Horse Maquerel	3.5%	5.3%	90.6%	0.4%	Horse Maquerel	8.8%	10.4%	80.3%	0.4%
Blue Whiting	0%	0%	3%	97%	Blue Whiting	7%	1%	13%	79%

Fig. 4 Confusion matrixes for the classifier that results from the combination of the discriminative model and the random forests. Confusion Matrixes are shown for the supervised learning (left) and the four-class mixture.

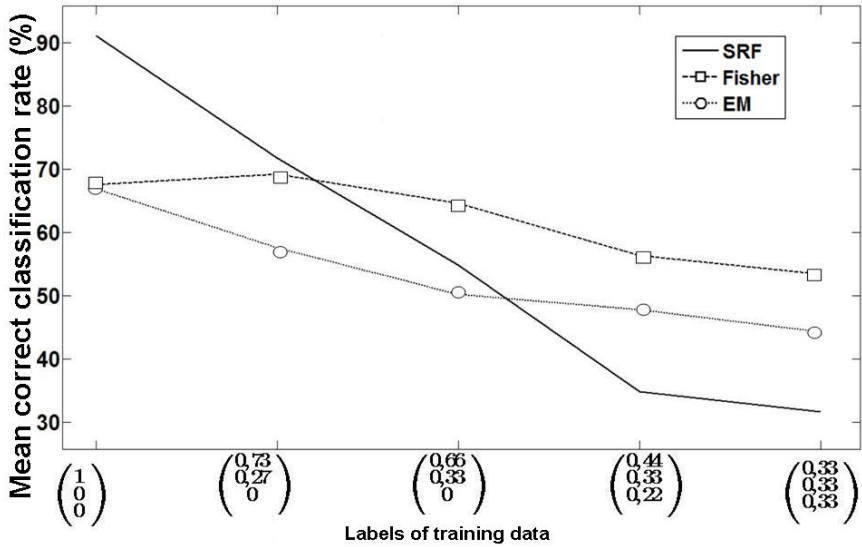


Fig. 5 Mean correct classification rate for 3-classes images with different target proportions going from the supervised case (on the left) to uniform situations (on the right)

be the less robust to the increased mixture complexity with a loss in classification performances greater than 50% between the supervised and unsupervised cases. In contrast, the performances of the discriminative models only decrease by less than 15%.

These additional experiments further validate the choice of the combination of the discriminative models and the random forest. It should be noted that for real applications training datasets would involve a variety of mixture complexities such

that the performances of the random forest would not be as degraded as in the extreme situations considered in figure 5. The combination of the two classifiers lead to the best results in all cases and the improvement w.r.t. random forests alone reach a classification gain up to 14% and 20%.

8 Conclusion

This paper is dedicated to weakly supervised learning. The majority of models processes training data that are labelled with binary vector indicating the presence or the absence of object class in images. Here training data are provided with prior labelling, the label being a vector that indicates the prior for each class. These training data are obtained with class proportion knowledge in images instead of presence/absence knowledge. This kind of training data is typical from fisheries acoustics that provide objects in images that are labelled with relative class proportion.

Three probabilistic classification models are presented and analysed. We intentionally choose models that are very different in terms of global and mathematical approaches: a generative model, a discriminative model and random forests. These three models take probabilities at the input and provide probabilities at the output. For the fisheries acoustics dataset, in supervised learning, random forests reach the better correct classification rate but results fall down in weakly supervised learning and are equivalent. The generative model provides the lower results with correct accuracy in supervised learning but very low performance in weakly supervised learning. The discriminative model is the more robust model as regard to the weakly supervised learning but accuracy is not correct. A classifier combination method has been then proposed to fuse two classification models and to combine their classification abilities, i.e. the strong accuracy and the robustness. Results prove the pertinence of the approach by providing more robust and accurate correct classification rates.

As regards to the application, the operational situations typically involve mixtures between two or three species and the reported recognition performances (between 90% and 77%) are relevant w.r.t. ecological objectives in terms of species biomass evaluation and the associated expected uncertainty levels. However, this approach does not take in account the spatial organisation of species in the given area. So, an effort must be done to include spatial information [16].

References

1. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by probability distributions. *Bull. Calcutta Maths. Soc.* 35, 99–109 (1943)
2. Breiman, L.: Random forests. *Machine Learning* 45, 5–32 (2001)
3. Breiman, L., Friedman, J., Olshen, R., Stone, C.: *Classification and regression trees*. Chapman and Hall, Boca Raton (1984)

4. Chapelle, O., Schölkopf, B., Zien, A.: *Semi-supervised learning*. MIT Press, Cambridge (2006)
5. Chung, J., Kim, T., Nam Chae, Y., Yang, H.: Unsupervised constellation model learning algorithm based on voting weight control for accurate face localization. *Pattern Recognition* 42(3), 322–333 (2009)
6. Crandall, D.J., Huttenlocher, D.P.: Weakly supervised learning of part-based spatial models for visual object recognition. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3951, pp. 16–29. Springer, Heidelberg (2006)
7. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. *Jour. of the RSS* 39, Series B(1), 1–38 (1977)
8. Dietterich, T.: An experimental comparison of three methods for constructing ensembles of decision trees. *Machine Learning* 40(2), 139–158 (2000)
9. Fablet, R., Lefort, R., Scalabrin, C., Massé, J., Boucher, J.M.: Weakly supervised learning using proportion based information: an application to fisheries acoustic. In: *International Conference on Pattern Recognition* (2008)
10. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale invariant learning. In: *Conference on Computer Vision and Pattern Recognition* (2003)
11. Fishburn, P.: *Utility theory for decision making*. John Wiley and Sons, New York (1970)
12. Fisher, R.: The use of multiple measurements in taxonomic problems. In: *Annals of Eugenics*, pp. 179–188 (1936)
13. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55, 119–139 (1997)
14. Gu, L., Xing, E., Kanade, T.: Learning gmrf structures for spatial priors. In: *Conference on Computer Vision and Pattern Recognition*, pp. 1–6 (2007)
15. Kass, G.: An exploratory technique for investigating large quantities of categorical data. *Journal of applied statistics* 29(2), 119–127 (1980)
16. Lefort, R., Fablet, R., Boucher, J.M.: Combining image-level and object-level inference for weakly supervised object recognition. application to fisheries acoustics. In: *International Conference on Image Processing* (2009)
17. Lefort, R., Fablet, R., Boucher, J.M.: Weakly supervised learning with decision trees applied to fisheries acoustics. In: *IEEE International Conference on Acoustics, Speech and Signal Processing* (2010)
18. Loh, W.Y., Shih, Y.Y.: Split selection methods for classification trees. *Statistica Sinica* 7, 815–840 (1997)
19. Petitgas, P., Massé, J., Beillois, P., Lebarbier, E., Le Cann, A.: Sampling variance of species identification in fisheries acoustic surveys based on automated procedures associating acoustic images and trawl hauls. *ICES Journal of Marine Science* 60(3), 437–445 (2003)
20. Ponce, J., Hebert, M., Schmid, C., Zisserman, A.: *Toward Category-Level Object Recognition*. LNCS. Springer, Heidelberg (2006)
21. Quinlan, J.: *C4.5: Programs for machine learning*. Morgan Kaufmann, San Francisco (1993)
22. Scalabrin, C., Massé, J.: Acoustic detection of the spatial and temporal distribution of fish shoals in the bay of biscay. *Aquatic Living Resources* 6, 269–283 (1993)
23. Schölkopf, B., Smola, A.: *Learning with Kernels*. The MIT Press, Cambridge (2002)
24. Schmid, C.: Weakly supervised learning of visual models and its application to content-based retrieval. *International Journal on Computer Vision* 56, 7–16 (2004)
25. Shivani, A., Roth, D.: Learning a sparse representation for object detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2353, pp. 113–127. Springer, Heidelberg (2002)

26. Ulusoy, I., Bishop, C.: Generative versus discriminative methods for object recognition. In: Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 258–265 (2005)
27. Vidal-Naquet, M., Ullmann, S.: Object recognition with informative features and linear classification. In: ICCV (2003)
28. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 18–32. Springer, Heidelberg (2000)
29. Xie, L., Perez, P.: Slightly supervised learning of part-based appearance models. In: Computer Vision and Pattern Recognition Workshop, vol. 6 (2004)

Intelligent Spaces as Assistive Environments: Visual Fall Detection Using an Evolutive Algorithm*

José María Cañas, Sara Marugán, Marta Marrón, and Juan C. García

Abstract. Artificial vision provides a remarkable good sensor when developing applications for intelligent spaces. Cameras are passive sensors that supply a great amount of information and are quite cheap. This chapter presents an application for elderly care that detects falls or faints and automatically triggers the health alarm. It promotes the independent lifestyle of elder people at their homes as the monitoring application will call for timely health assistance when needed. The system extracts 3D information from several cameras and performs 3D tracking of the people in the intelligent space. One evolutive multimodal algorithm has been developed to continuously estimate the 3D positions in real time of several persons moving in the monitored area. It is based on 3D points and learns the visual appearance of the persons and uses colour and movement as tracking cues. The system has been validated with some experiments in different real environments.

Keywords: detection, vision, fall, three-dimensional, eldercare.

1 Introduction

The aging of population all around the world, especially in Europe, challenges to economies and societies and also generates new needs, both at societal and individual levels. The elder sector of population is growing and technology

José María Cañas · Sara Marugán
Universidad Rey Juan Carlos
e-mail: jmplaza@gsyc.es, smarugan@gsyc.es

Marta Marrón · Juan C. García
Universidad de Alcalá
e-mail: marta@depeca.uah.es, jcarlos@depeca.uah.es

* This work has been partially funded by the Spanish Ministerio de Educación (DPI2007-66556-C03) and by the Comunidad Autónoma de Madrid(S2009/DPI-1559).

may play a big role improving their quality of life and serving both as an assistant and as an integration tool.

Advances in miniaturization of computing devices, new sensor elements and networking make it possible to embed some kind of computational intelligence into working environments, private homes or public spaces. The so called Intelligent Spaces open a wide range of possibilities of interaction between humans and the surrounding environment. Based on Intelligent Spaces resources and concepts, Assistive Technologies research has received a new impulse looking for new solutions and applications. Many different aspects of intelligent spaces concerning Assistive Technologies are open to research and development: sensor and monitoring devices, processing units and actuators, man-machine interaction, and even ethic or legal implications of the deployment of such systems.

Over one-third of elders 65-years-old fall each year [12]. The falls usually result in serious injuries like hip fracture, head traumas, etc. The rapid health assistance in case of fall may reduce the severity of the injuries. The care of elderly implies a continuous monitoring of their daily tasks. In many cases their own families or the social services are in charge of their care at their own homes or in specialized institutions. But even counting with the necessary amount of caregivers, it is impossible to watch these patients continuously in order to detect any incident as fast as possible. The problem worsens for people who live alone at home, as they need much more this type of assistance in case of emergency. An interesting application for elder care is to detect falls or faints in order to automatically trigger a health alarm. Such application would promote the independent lifestyle of elder people at their homes as the monitoring application will call for timely health assistance when needed.

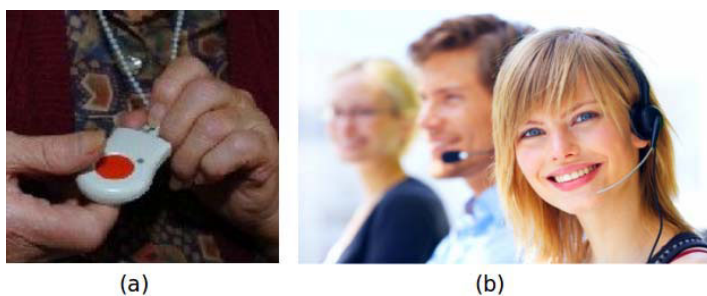


Fig. 1 Traditional tele-assistance system - (a) Pushing necklace, (b) Assistive service

In the context of fall detection and prevention there are several technological products in the market. First, traditional monitoring systems as pendants or wristbands worn by the patients [3], who must activate such devices when needed, usually pressing a button (Figure 1). The system sends an emergency

call to the appropriate health service. These traditional systems require human intervention to report an alarm or ask for help, and user's potential non-compliance (both intended and unintended) is a potential problem. In certain situations, for instance a faint that causes a fall to the floor, it will not be possible for the patient to activate the device, and that can be dangerous as the severity of the damage may increase with the time at the floor without health assistance. A second group of wearable systems relies on accelerometers and tilt sensors to automatically detect the falls [15]. Carrying this devices continuously may become a nuisance for the users.

Other solutions are embedded in the environment, they use external monitoring devices and then, the user's compliance is not required. There are systems which are based on floor-vibrations [1], on infrared array detectors [14] and on cameras. Inside this broad area of possibilities, artificial vision provides a remarkable good sensor when developing applications for intelligent spaces. Cameras are passive sensors that supply a great amount of information and most of them are quite cost effective. Several vision based assistive systems use omni-directional cameras [16, 9]. In particular [9] looks for activity patterns, models the patient's behavior and detects abnormal activities. Other works use optic flow as the main visual feature [5] or the motion history and human shape variation [13].



Fig. 2 Camera proliferation

One naive alternative is to use vision for external monitoring. A set of cameras transmit images to a service center where someone watches the display and decides whether a dangerous situation has been happened or not. This kind of *tele-watch systems* work in case of conscience loss, but have some disadvantages like the need of a person continuously watching the images and, even worse, it is inconvenient from the patient privacy. One way to overcome such disadvantages is to build *autonomous* tele-assistance systems, that continuously monitor the images without the need of a watching person and only show the patient images in case of alarm. The major difficulties building such systems lie in the complexity of extracting relevant information from the raw pixels in the image flow, in real time, and in the high computational cost it might demand. This kind of vision-based autonomous tele-assistance system shares a common background with general visual tracking techniques [4, 10, 11, 17], which have been applied to different scenarios.

In this chapter we present an autonomous tele-assistance system that tracks people positions in an Intelligent Space using a set of regular cameras at the same time. The system works in real time. When it detects some anomalous patient behavior, such as a falling to the floor, the system automatically can send an emergency message for immediate health assistance.

The system core is based on a novel evolutive multimodal algorithm which allows to continuously estimate the 3D position in real time and to learn the visual appearance of those people located into the covered area. People movement is described as a sequence of 3D positions. With this information, it is not difficult to determine whether people are laid on the floor, inside a dangerous area, etc.. The system has been validated with a set of experiments in different simulated and real environments.

The remainder of the chapter is organized in four additional sections. A global functional description of the fall detector system is presented in section 2. In section 3 the 3D localization algorithm and the tracking technology are described. In section 4 we show experimental results that validate and describe the system performance. Finally, conclusions are exposed in section 5.

2 Global System Description

For monitoring applications, a great part of the useful information in the work space is mainly three-dimensional, like the relative position of an object opposed to another or the movement of a person. One of the main problems in the identification of dangerous situations using vision sensors is their two dimension nature. For instance, when using a flat image to detect whether a person is near to an ignited oven, a window, a door, etc or not, there is ambiguity in the estimation of the distance, so we could easily make a mistake. High risk situations are better described, and in a more simple way, in 3D spatial terms.

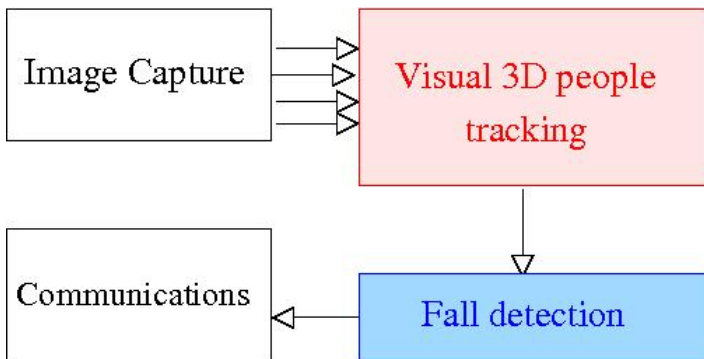


Fig. 3 Blocks diagram of ElderCare application

We have built an application, named *ElderCare*, whose main blocks are depicted at Fig. 3. First, the *image capture* block is responsible of getting the frames from several camera sensors along the monitored area. Analog cameras, wireless cameras, firewire and regular USB cameras are supported. Second, the *visual 3D people tracking* block extracts three-dimensional information in real time from the images, tracking the 3D position of every person at the monitored zone. It provides the current 3D position of every person at the area to the *fall detection* block. This third block defines a set of alarm rules which take into account 3D position and time conditions to trigger a health assistance alarm. For instance, if the position of a person is close to the floor (less than 20 cm) for a minute or more then the fall condition is triggered and an alarm is signaled to the *communications* block. This fourth block is responsible of sending such alarm to the health services via SMS, MMS, automatic phone call, etc.

Input data to the system consists of an image set of one or more house rooms and a set of rules that determine dangerous situations. For correct operation of the system there should be at least two cameras for each monitored room. Output data is an alarm signal sent, in general, to health assistant services or patient relatives. The system triggers it when detects a dangerous situation according to regular person positions, following certain rules. The rules that trigger the alarm are fully configurable and provide flexibility to the system in terms of alarm definition. They can be introduced during installation process.

In addition, a graphical interface has been developed, but only for debugging purposes (Figure 14). The system itself presents no window at operation time and records no single image to keep privacy of the monitored people. It has been implemented with a set of low cost cameras and a conventional PC. One 3D estimation technique has been carefully designed to run at real time on commodity hardware. It uses colour and movement to track the people 3D position. Combination of colour and movement offers some advantages over using each feature separately. On the one hand, motion locates and tracks people during their walk through the scene. It allows to reject all the static objects in the room and focus the visual computation on the regions of interest. On the other hand, colour information is easy to use and very selective. The combination of both features allows to learn colour when someone is moving and to keep her tracking when she is still. The algorithm will be described in detail in section 3.

3 Multimodal Evolutive Algorithm for Vision Based 3D Tracking

Evolutive algorithms manage an individual population that combine their properties to achieve new generations that approximate to the problem

solution. Normally evolutive algorithms execute repeatedly two steps until the populations converge to problem solution.

- New population generation.
- Fitness or quality computation for each individual.

Population generation requires genetic operators like *random mutation*, *thermal-noise*, *cross*, *repulsion* and others, it depends on the problem.

When an individual is compatible with sensorial observations its fitness will be high and also its probability for keeping it in the following generation. On the other hand, individuals not compatible with observations will have low fitness.

In this system, and individual is a 3D point, $P(x,y,z)$, due to the problem consists of finding the solution for 3D position estimation of people. The fitness is computed base on image information, specifically colour and motion.

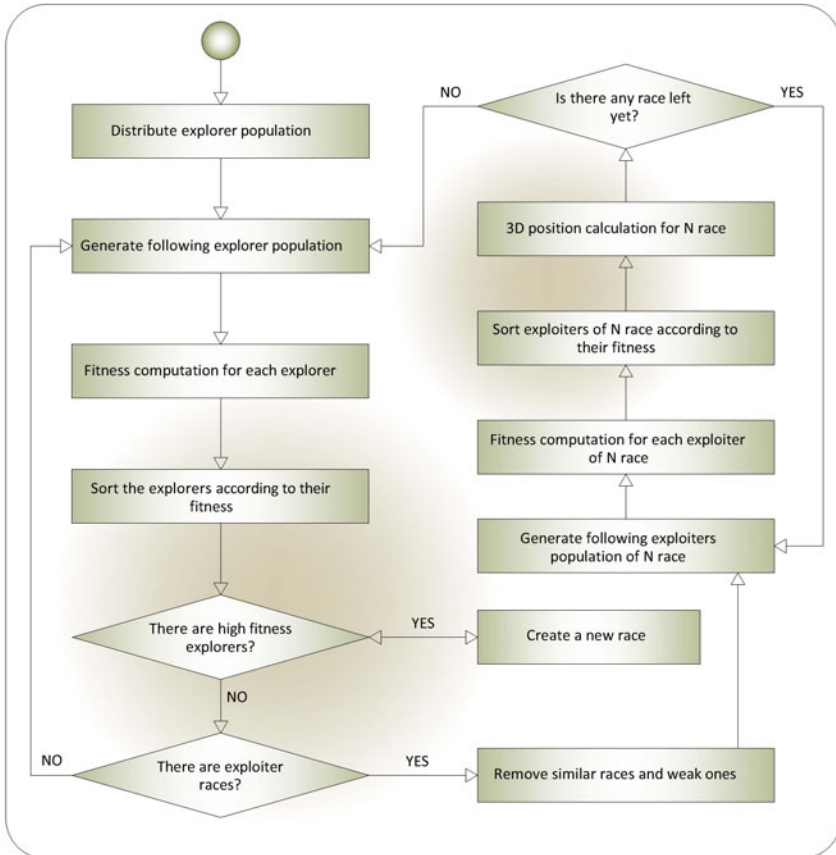


Fig. 4 Flow diagram

This is similar to other approaches with individuals in 3D space and fitness functions which are based on image features [6, 7, 8, 2].

The image 4 shows the flow diagram of the evolutive algorithm.

First of all we introduce two kinds of populations, then we describe how each kind of individual is evaluated through the fitness function and at least, we explain 3D position calculation.

3.1 Explorers and Races

Two types of populations are used: *explorers* and *exploiters*. Explorers search for movement in the whole space covered by cameras and it has a defined number of individuals. On the contrary, exploiters consist of an undefined number of races, each one analyzing one region where explorers have detected movement.

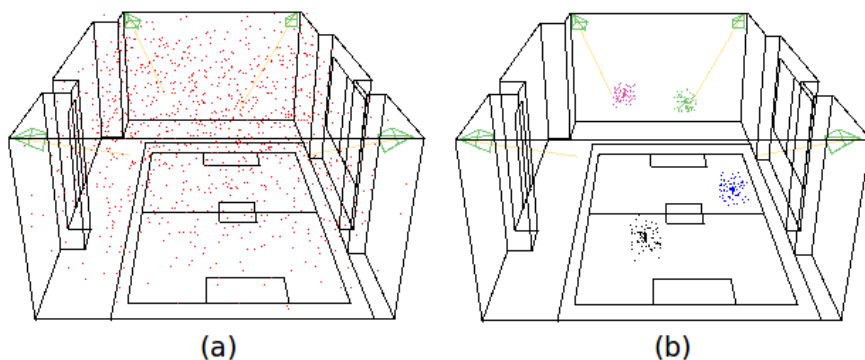


Fig. 5 Two types of populations: (a) Explorers, (b) Exploiters

To generate new populations the algorithm uses different genetic operators for each kind of population. Explorers are generated through *random mutation* and *abduction*. Random mutation consists of changing individual position randomly considering all position states inside the room. Abduction operator consists of generating new individuals in high probability space zones based on observations, zones where there is an object in movement. Abductions speed up exploiters solution convergence. The image 6(a) shows a person moving his arm and in the image 6(b) we can see the majority explorer population situated in compatible zones with movement in images.

On the other hand, exploiters are generated by elitism and thermal-noise operators. *Thermal-noise* operator is quite similar to random mutation, the difference is that thermal-noise explores new positions locally. The new individual position is near its old position.

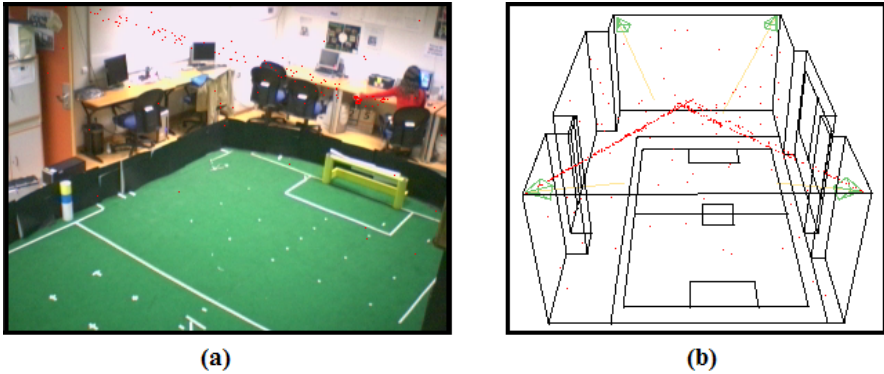


Fig. 6 Abduction operator: (a) Real image, (b) Virtual image

Elitism operator consists of passing the high fitness individuals without change to the next population. This allows to remember the best positions from the last iteration and to generate the thermal-noise individuals around them.

Depending on percentage of individuals assigned to each operator, exploiters behavior may vary. High thermal-noise configuration provokes that more exploiters explore the local zone.

The image 7(a) shows high percentage for thermal-noise and 7(b) image shows high percentage for elitism.

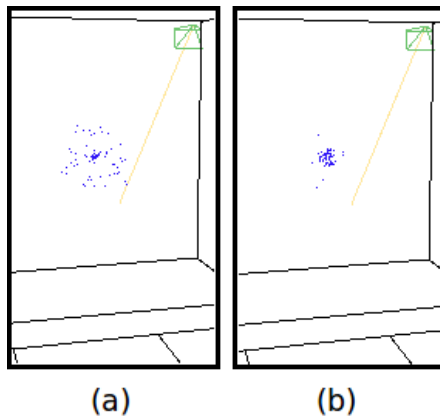


Fig. 7 Percentage configuration for thermal-noise and elitism: (a) High thermal-noise, (b) High elitism

3.2 Fitness Function Observation Model

Fitness calculation also requires a distinction between two types of populations. Explorer's fitness is calculated based on motion information. A high explorer's fitness is the trigger for tracker generation. Once one tracker is initiated, exploiters learn automatically the person clothes colour. Thus, fitness equations will be:

Explorer:

$$h_i = \sum P(\text{mov}_i | \text{img}_m) \quad (1)$$

Exploiter:

$$h_i = \frac{\sum P(\text{colour}_i | \text{img}_m) + \sum P(\text{mov}_i | \text{img}_m)}{2} \quad (2)$$

$$(3)$$

where $P(\text{colour}_i | \text{img}_m)$ and $P(\text{mov}_i | \text{img}_m)$ are m image pixel percent compatible with colour and movement.

For each individual, it calculates the pixel in that projects in each image and for fitness calculation also pixels in 5x5 neighborhood are analyzed. Pixel percent will be:

$$\sum P(\text{colour}_i | \text{img}_m) = \frac{k}{25} \sum P(\text{mov}_i | \text{img}_m) = \frac{k}{25} \quad (4)$$

where k is pixels that pass applied filter.

The algorithm generates a exploit race for each person in movement. A race tracks a person while he or she is visualized in two cameras at least. Two or more cameras are needed to extract three-dimensional information. They have to be calibrated in relation to each other.

Once the person is located inside the room, it can determine if there is a dangerous situation. The system detects if the person is fallen in the floor checking Z coordinate. If Z coordinate is under a threshold it activates a visual and audible alarm.

As we have mentioned before, this system extract colour and motion information from images. The following sections describe this process.

3.2.1 Motion Detection

When an object is in movement, some pixels changes their values. The system detects this change through consecutive frame comparison and background

extraction. The background extraction is calculated using a learned background image for each camera. These images are created as a result of a frame weighted sum in defined intervals (see equation 5).

$$background(t) = \alpha \times background(t - \beta) + (1 - \alpha) \times frame(t) \quad (5)$$

where alpha is in range $[0,1]$ and beta indicates time interval for background updating.

Frames comparison is a simply absolute difference between two images. If a pixel difference is above a defined threshold, this pixel passes the filter.

Thus, motion filter let pass pixels that have a significant difference with regard to previous frame or background image. The difference is determined between RGB images. The difference equation on a pixel $p(x,y)$ is:

$$DiffImg(x, y) = (diff_R \text{ AND } diff_G) \text{ OR } (diff_G \text{ AND } diff_B) \text{ OR } (diff_R \text{ AND } diff_B) \quad (6)$$



Fig. 8 Background learning: (a) Instantaneous image, (b) Background image

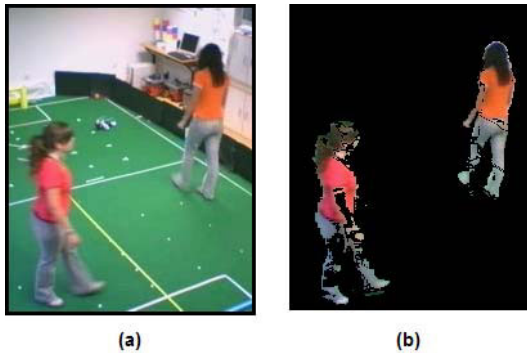


Fig. 9 Motion filter: (a) Real image, (b) Filtered image

where $diff_R$, $diff_G$ and $diff_B$ are booleans and indicates if the difference is above a given threshold. In order to consider real difference between pixels requires differences in at least two RGB channels.

Motion mask is calculated following this equation:

$$mask(x, y) = DiffImg_{prev}(x, y) \text{ OR } DiffImg_{backgd}(x, y) \quad (7)$$

where $DiffImg_{prev}$ is the binary image from previous image difference and $DiffImg_{backgd}$ from background image difference.

Then, fitness function uses motion mask for its equations.

3.2.2 Automatic Colour Learning

Object tracking only based on movement detection does not solve the problem when the object is still. Eldercare system learns clothe people colour to keep the track in this situation. Therefore, the system associates one colour to each person.

Firstly, the system uses motion filter to learn people colour and generates a colour filter dynamically for each race. It takes motion pixels as samples for a HSV histogram, that is associated to the race. Thus, the colour model used is Hue-Saturation-Value model.

HSV (Hue, Saturation and Value) - defines a type of colour space. It is similar to the modern RGB and CMYK models. The HSV colour space has three components: hue, saturation and value. 'Value' is sometimes substituted with 'brightness' and then it is known as HSB. The HSV model was created by Alvy Ray Smith in 1978. HSV is also known as the hex-cone colour model.e

- Hue represents colour. In this model, hue is an angle from 0 degrees to 360 degrees.
- Saturation indicates the range of grey in the colour space. It ranges from 0 to 100%. Sometimes the value is calculated from 0 to 1. When the value is '0,' the colour is grey and when the value is '1,' the colour is a primary colour. A faded colour is due to a lower saturation level, which means the colour contains more grey.
- Value is the brightness of the colour and varies with colour saturation. It ranges from 0 to 100%. When the value is '0' the colour space will be totally black. With the increase in the value, the colour space brightness up and shows various colours.

Filter consists of defining tolerances on each channel using as reference values the major HSV value registered by the histogram. A pixel passes the filter if has low HSV difference with respect to tolerances. The colour filter equation for a pixel i :

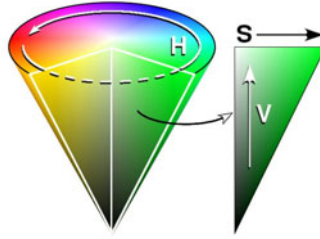


Fig. 10 HSV colour cone

$$Filter_i = abs(H - h_i)^1 < Htol \text{ AND } abs(S - s_i) < Stol \text{ AND } abs(V - v_i) < Vtol \quad (8)$$

where Htol, Stol and Vtol are filter tolerances.

The HSV filter allows to learn all kind of colours, including dark and pale colours (see image 11).

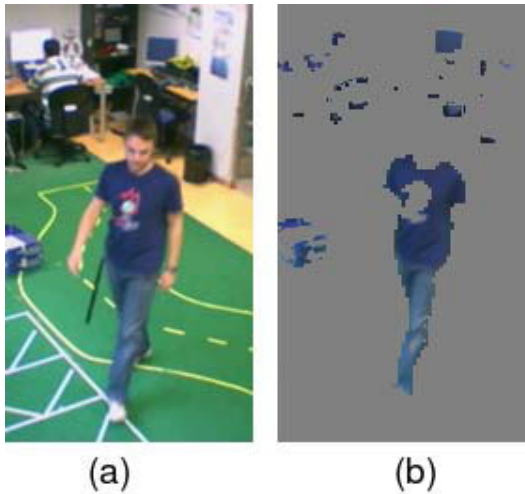


Fig. 11 Colour filter example: (a) Original image, (b) Filtered image

3.3 Determine 3D Positions

Three-dimensional position estimation is calculated through exploiter elitism individuals of each race with a weighted sum. Therefore, we have one position

¹ Angle subtraction.

estimation for each person inside the room. The reason for using only elitism individuals is the stability of the race position along the time.

$$race_{position} = \frac{\sum_{n=1}^N totalFitness_n * xyz_n}{N} \quad (9)$$

where N is elitism individuals number, $totalFitness_n$ is an elitism individual fitness in range $[0,1]$ and xyz_n is the 3D point that represents an elitism individual.

4 Experiments

We have done some experiments in the Robotics Laboratory in our University. This room has approximately 40 square meters and 4 cameras connected with local net and situated in the room up-corners. We use the *Jderobot*² platform developed in our Robotics Group. This is a middleware for generating autonomous behavior related to robotic, domotic and computer vision applications.

Jderobot has an active community of developers, so latest devices are supported. It is written in the C language that is a good compromise between power and efficiency. *Jderobot* has a human interface for managing applications. This applications in this architecture are called *schemas*. *Schemas* can be organized in hierarchy. For example, each one is in charge of a certain behavior and all together combined make up and the complete autonomous behavior.

The platform also offers several drivers and application examples (schemas) that use them. Eldercare uses three drivers:

- *Firewire driver*. This driver uses the libdc1394 library for getting images from firewire cameras. Cameras are iSight models and provide 30 fps.
- *Networkserver driver*. This driver serves the images through the net. Firstly receives the images from Firewire driver and then puts them on the net.
- *Networkclient driver*. Networkserver and Networkclient drivers have their own communication protocol to transmit the images. Networkclient driver receives images by main computer from four networkserver driver instances, one for each camera.

The main computer executes the algorithm and its characteristics are Intel Quad Core with 4GB RAM memory.

² <http://jderobot.org>



Fig. 12 Testing scenario - (a) iSight camera, (b) laboratory

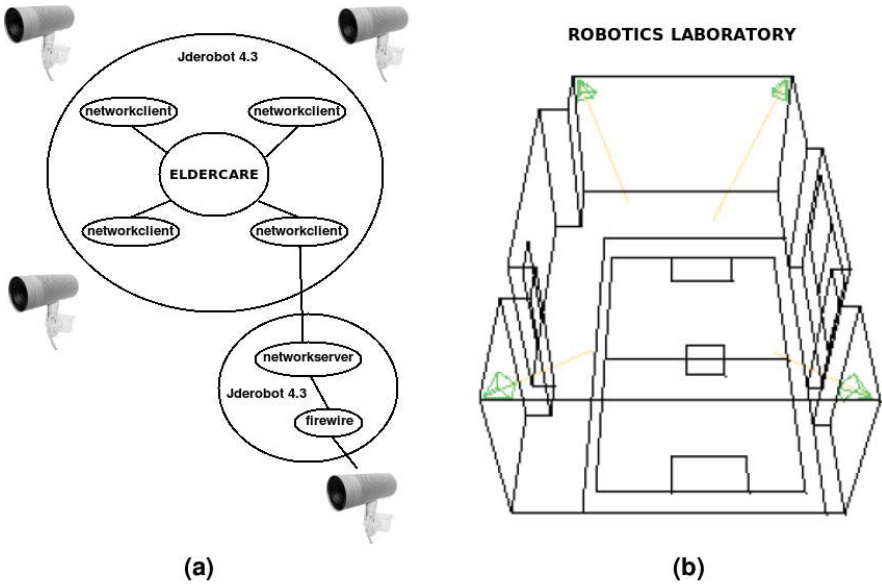


Fig. 13 Infrastructure diagram

Here we have an infrastructure diagram that shows how the application is connected to four networkserver driver instances executing in four different computers.

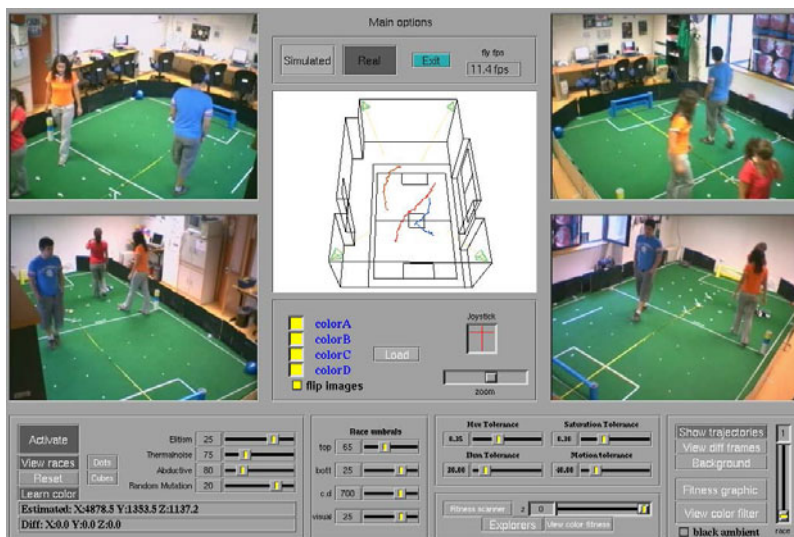


Fig. 14 Eldercare graphic interface

Another important element from our application is the graphic interface (see Fig. 14), that offers us the possibility to see the results of the algorithm and debug it.

4.1 Typical Execution

This experiment has consisted of tracking several people during a typical Eldercare execution. Switch on the system, some people enter inside the room and the system tracks them without difficulty keeping 30 iterations per second. In fact, the algorithm can reach 55 fps but image flow works on 30 fps. Due to this restriction, we force the application to keep the same frame ratio. Therefore, the system works with sufficient required speed.

In the images below we can see a typical execution for tracking people. This experiment shows us that the system can track more than one person without problem.

In the first image we can see the trajectory that has followed the person since she entered the room until she is standing up in the middle of the room. The system can obtain 3D person position repeatedly in real time.

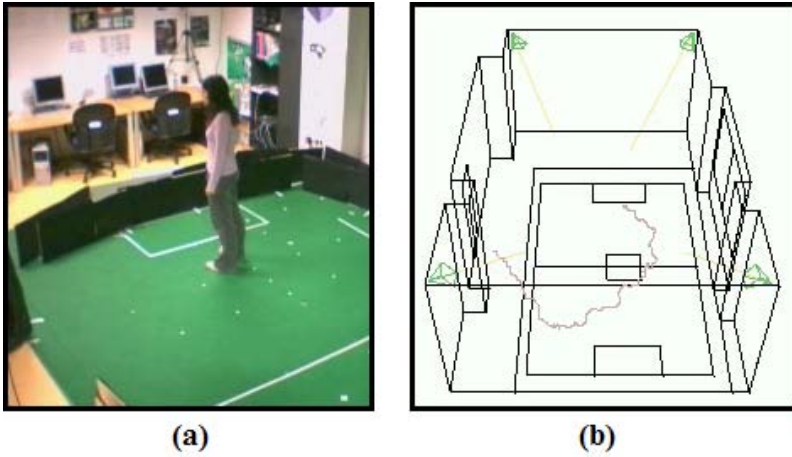


Fig. 15 One person tracking

In the second image two people appear that Eldercare is tracking and their repetitive trajectories. We checked that the system can track two people without losing real time performance.

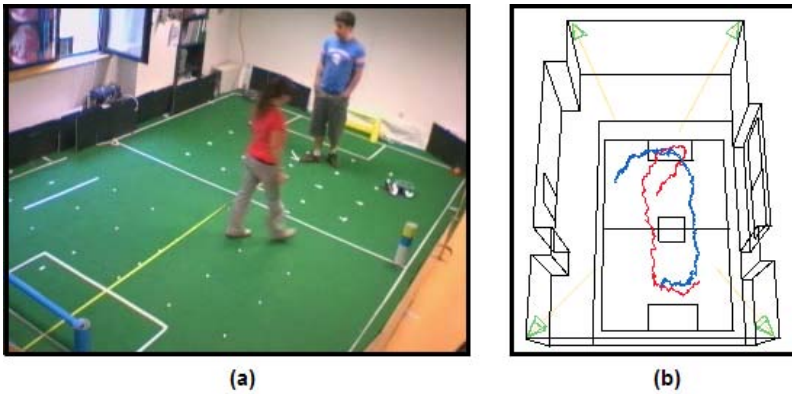


Fig. 16 Two people tracking

The last image shows three people tracking. In the right image above there are races drawn through all their individuals, like a cloud. Tracking three people the system works on 26 fps.

Another experiment consists of tracking one person that falls on the floor. In this situation, 3D position is too near to the floor and the system activates a visual and audible signal. The Z coordinate threshold that we choose is 30 centimeters. It allows to detect a person laying on the floor in all cases.

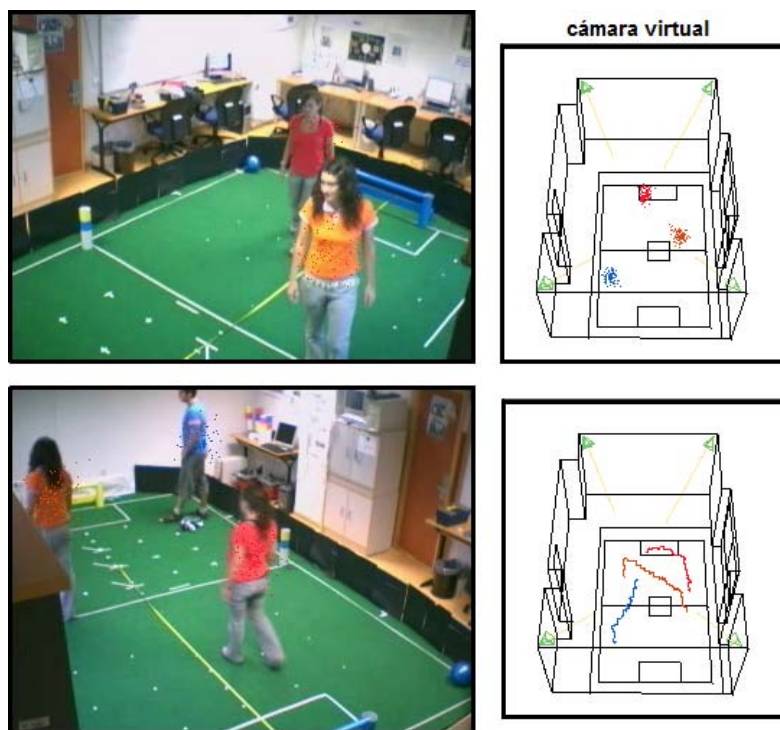


Fig. 17 Three people tracking

4.2 Time Performance

In this experiment we have analyzed Eldercare time performance in two different terms.

For the first term we used *Oprofile*, a continuous system-wide profiler for Linux. It consists of a kernel driver and a daemon for collecting sample data, and several post-profiling tools for turning data into information. With this tool we know that in a typical execution 85% of the samples belong to Eldercare source code. In particular, we obtain sample percentages, that mean time percentages, for the main functions of the algorithm (see table 4.2).

The second kind of data consists of measuring time intervals for the same main functions in milliseconds. This gives us a good idea about time requirements.

In the table we can see three groups of functions. The first one contains functions for checking and generating algorithm populations. Regarding race similarity check function, it takes 5 milliseconds only if there are more than 5 races.

The second group belongs to fitness function, that takes the major part of the algorithm time. The motion complete filter and the individual window



Fig. 18 Detected fall

filter are called from this function. For each exploiter individual the algorithm executes a window filter for each instantaneous camera image. Fitness calculation takes 2 milliseconds for exploiter population and 13 milliseconds for explorer population.

The last group are visualization functions to show the results on the GUI.

Functions	Time %	Milliseconds
Race similarity check	0.17	0 - 5
New population generation	3.24	1
Fitness calculation (FC)	69.03	2 - 13
FC - complete image motion filter	42.33	9
FC - individual window image colour filter	5.03	4
Virtual image generation (GUI thread)	1.28	5
Fill visualization buffers (GUI thread)	9.14	5

4.3 System Accuracy

To analyze the system accuracy we have done two kinds of experiments. The first one consists of comparing simulated 3D object positions to positions estimated by the algorithm. *Jderobot* offers a driver that generate virtual images called *Simulated3D*. This driver receives a world configuration file and camera parameters to simulate schematic images from camera's point of view. Here we have a diagram that represents this Eldercare configuration.

During eight seconds we have been collecting distance errors between simulated object position and estimated position from algorithm. Then we have

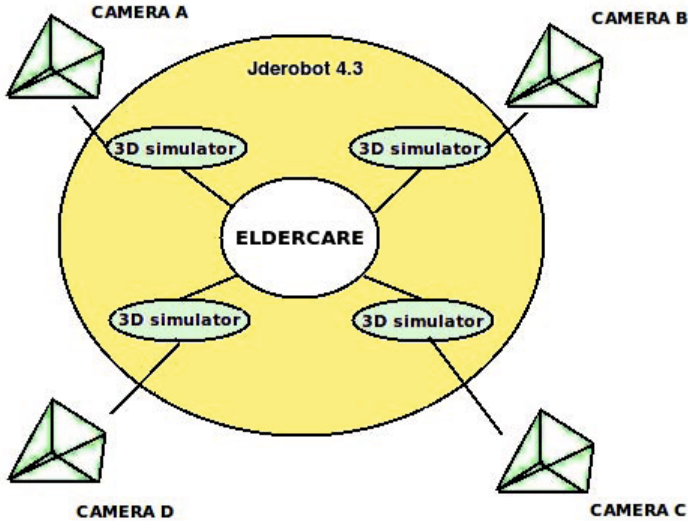


Fig. 19 Eldercare connected to simulated3D driver

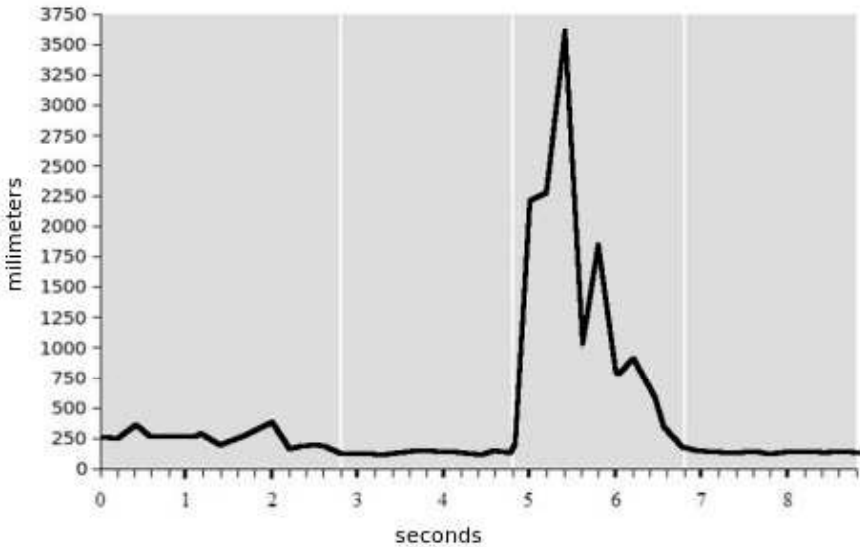


Fig. 20 Position estimation error graphic for simulated 3D object

put the results on the graphic below. This experiment shows us the system position estimation error without calibration camera errors and well-known object 3D positions.

For real 3D positions we measured 3D person position in different points of the room, considering the middle of the torso as the person position. We

have selected five 3D point (see image 21) that are visible by all the cameras. We measured the estimation error for each point and the tables below show the results.

- Using four cameras.

Point	X error	Y error	Z error	Total error (mm)
P1	118	148	91	210
P2	136	127	111	217
P3	80	160	75	194
P4	120	70	95	168
P5	94	88	158	204

- Using three cameras.

Point	X error	Y error	Z error	Total error (mm)
P1	108	56	143	188
P2	119	130	123	215
P3	148	86	137	220
P4	138	159	102	234
P5	83	122	129	197

- Using two cameras from the same side of the room (AB from image 21).

Point	X error	Y error	Z error	Total error (mm)
P1	145	130	147	244
P2	158	207	120	287
P3	145	167	213	308
P4	154	235	165	326
P5	214	174	178	328

- Using two cameras from the same side of the room (AD from image 21).

Point	X error	Y error	Z error	Total error (mm)
P1	207	239	224	388
P2	198	286	254	431
P3	268	304	209	456
P4	305	249	300	496
P5	226	308	287	478

Mean estimation errors:

Number of cameras	Mean error (mm)
4 cameras	199
3 cameras	211
2 cameras (AB)	299
2 cameras (AD)	450

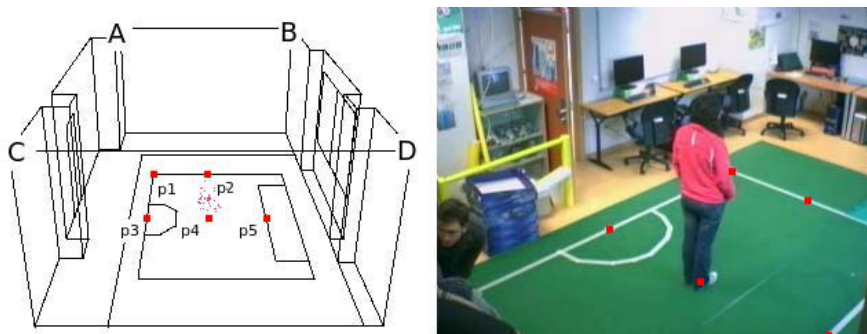


Fig. 21 Position estimation error experiment

The results show that Eldercare position estimation error with three or more cameras is in the order of 10 centimeters. This permits the system to differentiate between a person sitting on the floor and laying on the floor.

The data also show that estimation error decreases introducing more cameras. Using three cameras we obtained similar results to using four. Nevertheless, using two cameras increases the estimation error in 10 centimeters if the cameras are at the same side of the room. With two cameras situated in different sides of the laboratory the error is quite high.

5 Conclusions

We have presented a system called Eldercare that detects automatically dangerous situations for tele-assistance of old people. It activates an alarm when these situations happen, specially when a person falls on the floor. The alarm may also transmit a signal through sending a message to assistance service.

The system is made up of several cameras, a personal computer and image processing software. The hardware is conventional and it permits an easy installation and maintenance. Software is based on advanced technology for visual 3D racking and genetic multiobjective algorithm.

We have built an experimental prototype and it has been proved detecting people falls and warning about dangerous situations through an acoustic alarm. The algorithm is fast enough to detect falls in the same instant that they happen, which permits a quick response to emergencies.

There are several future lines to improve the current system. The first one consists of changing the individual primitive from 3D point to a prism, that has volume. People will be better represented and possibly position estimation error will decrease. A second way for improving Eldercare is managing

alarms from cell phones. The system would be more useful if a relative from the patient family may receive health alarms in her cell phone about the patient situation.

References

1. Alwan, M., Rajendran, P.J., Kell, S., Mack, D., Dalal, S., Wolfe, M., Felder, R.: A Smart and Passive Floor-Vibration Based Fall Detector for Elderly. In: The 2nd IEEE International Conference on Information & Communication Technologies: from Theory to Applications - ICTTA 2006, April 24 - 28, Damascus, Syria (2006)
2. Barrera, P., Cañas, J., Matellán, V.: Visual object tracking in 3D with color based particle filter. *Int. Journal of Information Technology* 2(1), 61–65 (2005)
3. Brownsell, S., Hawley, M.S.: Fall monitoring. In: Wootton, R., Dimmicky, S., Kvedar, J. (eds.) *Home telehealth: connecting care within the community*, pp. 108–120. Royal Society of Medicine Press (2006)
4. Fritsch, J., Kleinhagenbrock, M., Lang, S., Fink, G., Sagerer, G.: Audiovisual person tracking with a mobile robot. In: *Proceedings of Int. Conf. on Intelligent Autonomous Systems*, pp. 898–906 (2004)
5. Huang, C.L., Chen, E.L., Chung, P.C.: Fall Detection using Modular Neural Networks and Back-projected Optical Flow. *Biomedical Engineering - Applications, Basis and Communications* 19(6), 415–424 (2007)
6. Louchet, J.: Stereo analysis using individual evolution strategy. In: *Proceedings of the 15th International Conference on Pattern Recognition*, pp. 908–911 (2001)
7. Louchet, J.: Using an individual evolution strategy for stereovision. *Genetic Programming and Evolvable Machines* (2001)
8. Louchet, J., Guyon, M., Lesot, M.J., Boumaza, A.: Dynamic flies: a new pattern recognition tool applied to stereo sequence processing. *Pattern recognition letters* (2002)
9. Miaou, S.G., Shih, F.C., Huang, C.Y.: A Smart Vision-based Human Fall Detection System for Telehealth Applications. In: Bashshur, R. (ed.) *Proc.Third ISATED Int. Conf. on Telehealth*, Montreal, Canada, pp. 7–12. Acta Press (2007)
10. Pérez, P., Vermaak, J., Blake, A.: Data fusion for visual tracking with particles. *Proceedings of IEEE* 92(3), 495–513 (2004)
11. Pupilli, M., Calway, A.: Real-Time Camera tracking using a particle filter. In: *Proceedings of British Machine Vision Conference*, pp. 519–528 (2005)
12. Rajendran, P., Corcoran, A., Kinoshian, B., Alwan, M.: Falls, Fall Prevention, and Fall Detection Technologies. In: *Eldercare Technology for Clinical Practitioners*, pp. 187–202. Humana Press (2008)
13. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Fall Detection from Human Shape and Motion History Using Video Surveillance. In: *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW 2007)*, vol. 2, pp. 875–880 (2007)
14. Sixmith, A., Johnson, N.: A smart sensor to detect the falls of the elderly. *Pervasive Computing* 3(2), 42–47 (2004)

15. Yang, C.C., Hsu, Y.L.: Developing a Wearable System for Real-Time Physical Activity Monitoring in a Home Environment. In: Bashshur, R. (ed.) Proc. Third ISATED Int. Conf. on Telehealth, Montreal, Canada. Acta Press (2007)
16. Yiping, T., Shunjing, T., Zhongyuan, Y., Sisi, Y.: Detection Elder Abnormal Activities by using omni-directional vision sensor: activity data collection and modeling. In: Proc. Int. Joint Conference SICE-ICASE, pp. 3850–3853 (2006)
17. Zotkin, D., Duraiswami, R., Davis, L.: Multimodal 3D tracking and event detection via the particle filter. In: IEEE Workshop on Detection and Recognition of Events in Video, pp. 20–27 (2001)

Author Index

- Bakar, Rohani Abu 75
Barriga, Angel 133
Boucher, Jean-Marc 203
Boylan, Geraldine B. 93
Cañas, José María 225
Dong, Fangyan 179
Fablet, Ronan 203
Ferreira, Pedro M. 21
García, Juan C. 225
Hassan, Nashaat Mohamed Hussein 133
Hirota, Kaoru 179
Iliyasu, Abdullahi M. 179
Lefort, Riwal 203
Le, Phuc Q. 179
Lightbody, Gordon 93
Marnane, William P. 93
Marrón, Marta 225
Marugán, Sara 225
Pedrycz, Witold 1
Ruano, António E. 21
Russo, Fabrizio 115
Temko, Andrey 93
Thomas, Eoin M. 93
Várkonyi-Kóczy, Annamária R. 155
Wang, Shuming 1
Watada, Junzo 1, 55, 75
Yaakob, Shamshul Bahar 55
Yu-Yi, Chu 75

Subject Index

3D model reconstruction 156, 157, 175
 automatic 156, 160
 image based 157, 175, 176

A

algorithm
 corner detection 156
 evolutionary 25
 Gaussian smoothing 158, 159
 Levenberg-Marquardt 30, 31
 sketch based image retrieval 166, 176
 training 23, 30, 31

B

B-spline network 30
basis function 32
Battacharrya distance 210
Boltzmann machine 56, 57, 59, 62, 66,
 67, 69, 72, 73
Boolean Expression 189
bounded
 product 133, 134, 142–145, 148, 153
 sum 133, 134, 141–144, 153

C

characteristic
 edge 156, 171, 172, 174
 information 156, 175
chromosome
 encoding 26
 representation 26

classification 22
 Maximum margin 100
 Kernel trick 100
 Bayesian classifier
 Classifier 213
 discriminative 209, 211
 generative 205
 combination 214
cloudiness estimation 43
clustering 75–79, 81–85, 89–91
Collateral distortion 129
combinatorial problem 23, 24
Complete-line 77
complexity reduction 156
confidence interval 3, 4, 9, 10, 11, 13, 15,
 17, 18
contrast control 133, 134, 136, 139,
 141–145, 148, 149, 152, 153
corner detector 156, 157, 159
 Förstner's 158, 160, 163, 165, 167
 fuzzy 160, 162, 163, 166, 175
 Harris 158, 163, 165, 168
 SUSAN 158, 163, 165, 169
Cross-validation (leave one out) 96
crossover 28
 identity preserving 28
curvature
 mean 169, 170, 173
 principal 170

D

Decision Tree 212
degree of cornerness 160, 162, 175

Detail preservation 124
 DNA computing 76
 double-layer 56, 59, 62, 63

E

early-stopping 34
 edge
 detection 169, 170
 map 171, 172, 174
 object contour 168
 separation 171
 texture 168
 edge extraction
 primary 169, 170, 175
 electricity load demand 35
 Electroencephalography (EEG) 94
 Background (neonatal) 94
 Seizures (neonatal) 94, 106
 entropy 134, 135, 139, 140
 Expectation Maximization 101
 expected value 1, 4, 5, 6, 8

F

feature extraction 155, 156, 159, 166, 167
 filters 116, 117
 FIRE 162
 Fisher 209
 Fisheries acoustics 215
 fitness assignment 27
 Flexible representation of quantum images
 180
 Frequency analysis
 PSD (power spectral density) 98
 FFT (fast fourier transform) 98
 Wavelets 98
 function
 approximation 22
 beneficial 160
 fuzzy
 Classes 123
 corner detector 160, 162, 163, 166, 175
 edge detector 169, 170, 175
 filters 159, 175
 inference 146, 148, 150, 152
 measure 175
 membership 123-127
 models 116

random regression model 3, 7, 9, 10
 random variable 3-6, 9
 regression model 2-4, 7-12
 Relations 122-127
 rule 146, 150-152
 Systems 116

G

Gaussian
 activation function 31
 convolution 160
 Distribution 100-102
 Mixture model (GMM) 101
 Noise 116
 smoothing 158, 159

H

Hamiltonian path problem 75
 hardware
 implementation 133, 134, 141, 142,
 152
 Hierarchical agglomerative algorithms 77
 histogram 134, 140, 142-145, 153
 Hjorth parameters 99
 Hopfield network 55-58

I

image
 based modeling 157
 content analysis 167
 Denoising
 Enhancement 155, 156
 ground-based all-sky 43
 Quality
 segmentation 44
 Sharpening 118
 thresholding 47
 understanding 163, 167
 image retrieval system
 text based 167
 information
 extraction 172, 176
 primary 166, 168, 172, 174, 176
 retrieval 163, 167
 Invertible transformation
 input features 24, 26

J

Jacobian matrix 32

K

Kernel trick 210
 k-means 77, 79, 84, 86
 K-median 77

L

Levenberg-Marquardt 30
 LDA (linear discriminant analysis) 102

M

Machine learning 203
 mating pool 28
 matrix
 Confusion 103
 Covariance 102
 Jacobian 32, 33
 local structure 156–157, 160, 175
 Perspective Projection 175
 Maximum likelihood
 mean-variance analysis 55, 59, 61, 66, 67,
 68, 69, 71
 Minimal spanning tree 77
 model 96
 AR (autoregressive) models 99
 design cycle 29
 fuzzy random regression 1, 3, 4, 7, 8, 9,
 10, 11
 fuzzy regression 1, 2, 3, 7, 8, 11
 identification 22
 image based 157
 initialisation
 Mixture models 101
 search space 29
 situational 156, 175, 176
 3D
 multi-layer perceptron 30
 multiobjective
 evolutionary algorithm 25
 genetic algorithm 38, 50
 optimisation 23, 25
 mutation 28

N

neural network
 artificial 22
 design problem 23
 feed-forward 26, 30
 parameter initialisation
 parameters 23–25
 structure 23, 24
 topology 23, 25, 26, 30
 training 31
 neuro-fuzzy network 30
 neuron 22, 32
 noise 134, 135, 140, 141, 143
 Amplification 116
 cancellation 156
 elimination 159, 175
 filtering, FIRE 162
 non-linear
 autoregressive 27
 autoregressive with exogenous inputs
 27
 least-squares 31
 mapping 22
 parameters 23, 31, 32
 Normal distribution
 NP-Complete problem 75

O

object recognition 167, 173, 174
 objective
 functions 24
 space 29
 optimisation problem 23
 overtraining 34

P

parameter separability 30
 Pareto
 front 25, 26, 41, 47
 point 25
 set 25, 41, 47
 pattern matching 22
 point correspondence matching 156, 160,
 162, 175
 fuzzy 156, 175
 prediction 35
 long-term 39

performance 39
 Type A 123
 Type B 126
 Polynomial Preparation theorem 180
 PR (precision Recall) curve 103
 Probability density function 100

Q

Quantum

Computation 179
 Information 179
 Computer 179
 Algorithm 179
 State 179
 Image 179
 Image Compression 179
 Signal processing transformation 180
 Qubit Lattice 180

R

radial basis function 30, 32
 Random Forest 30
 Real Ket 30, 50
 recombination 28
 Residual noise 203
 robustness 180
 ROC (receiver operating characteristic)
 curve 103

S

search space 25
 sensor 133–135, 153

Shannon entropy 99
 Single-line 77
 structural learning 77
 surface 55
 deformation 169–173
 smoothing 169
 SVD (singular value decomposition) 99
 SVM (support vector machine) 96, 100

T

termination criteria 34
 training criterion
 modified 31
 standard 32

U

Unsharp masking
 Linear 115
 Nonlinear 116
 useful information extraction 156, 163,
 166, 174

V

variance 1, 3, 5, 6, 9, 17
 VHDL 142

W

wavelet network 30
 Weakly supervised learning 203