

Bartłomiej Beliczynski  
Andrzej Dzieliński  
Marcin Iwanowski  
Bernardete Ribeiro (Eds.)

LNCS 4432

# Adaptive and Natural Computing Algorithms

8th International Conference, ICANNGA 2007  
Warsaw, Poland, April 2007  
Proceedings, Part II

2  
Part II

 Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz

*University of Bern, Switzerland*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*University of Dortmund, Germany*

Madhu Sudan

*Massachusetts Institute of Technology, MA, USA*

Demetri Terzopoulos

*University of California, Los Angeles, CA, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Moshe Y. Vardi

*Rice University, Houston, TX, USA*

Gerhard Weikum

*Max-Planck Institute of Computer Science, Saarbruecken, Germany*

Bartłomiej Beliczynski Andrzej Dzielinski  
Marcin Iwanowski Bernardete Ribeiro (Eds.)

# Adaptive and Natural Computing Algorithms

8th International Conference, ICANNGA 2007  
Warsaw, Poland, April 11-14, 2007  
Proceedings, Part II

## Volume Editors

Bartłomiej Beliczynski  
Andrzej Dzielinski  
Marcin Iwanowski  
Warsaw University of Technology  
Institute of Control and Industrial Electronics  
ul. Koszykowa 75, 00-662 Warszawa, Poland  
E-mail: {B.Beliczynski,A.Dzielinski,M.Iwanowski}@ee.pw.edu.pl

Bernardete Ribeiro  
University of Coimbra  
Department of Informatics Engineering  
Polo II, 3030-290 Coimbra, Portugal  
E-mail: bribeiro@dei.uc.pt

Library of Congress Control Number: 2007923870

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

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

ISSN           0302-9743  
ISBN-10       3-540-71590-8 Springer Berlin Heidelberg New York  
ISBN-13       978-3-540-71590-0 Springer Berlin Heidelberg New York

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, re-use of illustrations, recitation, broadcasting, reproduction on microfilms 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.

Springer is a part of Springer Science+Business Media

[springer.com](http://springer.com)

© Springer-Verlag Berlin Heidelberg 2007  
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India  
Printed on acid-free paper   SPIN: 12041145   06/3180   5 4 3 2 1 0

# Preface

The ICANNGA series of conferences has been organized since 1993 and has a long history of promoting the principles and understanding of computational intelligence paradigms within the scientific community. Starting in Innsbruck, in Austria (1993), then Ales in France (1995), Norwich in England (1997), Portoroz in Slovenia (1999), Prague in Czech Republic (2001), Roanne in France (2003) and finally Coimbra in Portugal (2005), the ICANNGA series has established itself as a reference for scientists and practitioners in this area. The series has also been of value to young researchers wishing both to extend their knowledge and experience and to meet experienced professionals in their fields.

In a rapidly advancing world, where technology and engineering change dramatically, new challenges in computer science compel us to broaden the conference scope in order to take into account new developments. Nevertheless, we have kept the acronym ICANNGA which, since the Coimbra conference in 2005, stands for International Conference on Adaptive and Natural Computing Algorithms.

The 2007 conference, the eighth in the ICANNGA series, took place at the Warsaw University of Technology in Poland, drawing on the experience of previous events and following the same general model, combining technical sessions, including plenary lectures by renowned scientists, with tutorials and workshop panels.

The Warsaw edition of ICANNGA attracted many scientists from all over the world. We received 474 mostly high-quality submissions from 40 countries. After rigorous review involving more than 160 experts in their fields, 178 papers were accepted and included in the proceedings. The acceptance rate was only 38%, enforcing a high standard of papers. The conference proceedings are published in two volumes of Springer's *Lecture Notes in Computer Science*.

The first volume of the proceedings is primarily concerned with issues related to various concepts and methods of optimization, evolutionary computations, genetic algorithms, particle swarm optimization, fuzzy and rough systems. Additionally there is also a set of papers devoted to clustering and classification. The second volume is mainly concerned with neural networks theory and applications, support vector machines, biomedical and biometrics applications, computer vision, control and robotics.

ICANNGA 2007 enjoyed plenary lectures presented by distinguished scientists: Shun-ichi Amari from Japan, Ryszard Tadeusiewicz and Janusz Kacprzyk from Poland, Kevin Warwick and Rafal Zbikowski from England.

We would like to thank the International Advisory Committee for their guidance, advice and discussions. Our special gratitude is devoted to the Program Committee and reviewers. They have done a wonderful job of shaping the conference image.

Camera-ready version of the papers were carefully examined and verified by Wiktor Malesza, Konrad Markowski, Tomasz Toczyski and Maciej Twardy. A number of people from our Electrical Engineering Faculty, the Control Division Staff members and the PhD students were involved in various conference tasks, supporting the conference secretariat and maintaining multimedia equipment. We greatly appreciate all they have done.

We also wish to thank our publisher, especially Alfred Hofmann the Editor-in-Chief of LNCS and Anna Kramer for their support and collaboration.

Finally, the conference was made up of papers and presentations prepared by our contributors and participants. Most of our gratitude is directed to them.

April 2007

Bartłomiej Beliczynski  
Andrzej Dzielinski  
Marcin Iwanowski  
Bernardete Ribeiro

# Organization

## Advisory Committee

Rudolf Albrecht, University of Innsbruck, Austria  
Andrej Dobnikar, University of Ljubljana, Slovenia  
Vera Kurkova, Academy of Sciences of the Czech Republic, Czech Republic  
David Pearson, University Jean Monnet, France  
Bernardete Ribeiro, University of Coimbra, Portugal  
Nigel Steele, Coventry University, UK

## Program Committee

Bartłomiej Beliczynski, Poland (Chair)	Vera Kurkova, Czech Republic
Rudolf Albrecht, Austria	Pedro Larranaga, Spain
Gabriela Andrejkova, Slovakia	Francesco Masulli, Italy
Paulo de Carvalho, Portugal	Leila Mokhnache, Algeria
Ernesto Costa, Portugal	Roman Neruda, Czech Republic
Andrej Dobnikar, Slovenia	Stanislaw Osowski, Poland
Marco Dorigo, Belgium	Nikola Pavesic, Slovenia
Antonio Dourado, Portugal	David Pearson, France
Gerard Dray, France	Maria Pietrzak-David, France
Andrzej Dzielinski, Poland	Colin Reeves, UK
Jorge Henriques, Portugal,	Bernardete Ribeiro, Portugal
Katerina Hlavackova-Schindler, Austria	Henrik Saxen, Finland
Osamu Hoshino, Japan	Marcello Sanguineti, Italy
Janusz Kacprzyk, Poland	Jiri Sima, Czech Republic
Tadeusz Kaczorek, Poland	Catarina Silva, Portugal
Paul C. Kainen, USA	Nigel Steele, UK
Helen Karatza, Greece	Miroslaw Swiercz, Poland
Miroslav Karny, Czech Republic	Ryszard Tadeusiewicz, Poland
Marian P.Kazmierkowski Poland	Tatiana Tambouratzis, Greece
Mario Koeppen, Germany	Kevin Warwick, UK
Jozef Korbicz, Poland	Stanislaw H. Zak, USA

## Organizing Committee

Bartłomiej Beliczynski (Chair)  
Bernardete Ribeiro (Past Chair)  
Witold Czajewski (Technical Support, Conference Events)  
Andrzej Dzielinski (Reviewing Process)  
Waldemar Graniszewski (Social Program)  
Marcin Iwanowski (Conference Coordinator; Proceedings, WWW)  
Grazyna Rabij (Finances)

## Reviewers

Rudolf Albrecht	Soowhan Han
Krzysztof Amborski	Zenon Hendzel
Gabriela Andrejkova	Jorge Henriques
Jaroslav Arabas	Mika Hirvensalo
Piotr Arabas	Katarina Hlavackova-Schindler
Prasanna Balaprakash	Osamu Hoshino
Bartłomiej Beliczynski	Yanhai Hu
Conrad Bielski	Ben Hutt
Fatih Mehmet Botsali	Naohiro Ishii
Cyril Brom	Marcin Iwanowski
Pawel Buczynski	Wojciech Jedruch
Paulo de Carvalho	Tatiana Jaworska
Hasan Huseyin Celik	Piotr Jedrzejowicz
Leszek Chmielewski	Sangbae Jeong
YoungSik Choi	Marcel Jirina
Michal Choras	Tomasz Kacprzak
Ryszard Choras	Janusz Kacprzyk
Gyo-Bum Chung	Tadeusz Kaczorek
Andrzej Cichocek	Paul C. Kainen
Ernesto Costa	Helen Karatza
David Coufal	Andrzej Karbowski
Boguslaw Cyganek	Ali Karci
Witold Czajewski	Miroslav Karny
Włodzimierz Dabrowski	Włodzimierz Kasprzak
Dariusz Krol	Marian P. Kazmierkowski
Guy De Tre	Adnan Khashman
Andrej Dobnikar	Chang-Soo Kim
Antonio Dourado	Il-Hwan Kim
Gerard Dray	Kwang-Baek Kim
Andrzej Dzielinski	Mi-Young Kim
Mehmet Onder Efe	Mario Koeppen
Maria Ganzha	Jozef Korbicz
Waldemar Graniszewski	Anna Korzynska



Jacek Kozak  
Wojciech Kozinski  
Marek Kowal  
Petra Kudova  
Piotr Kulczycki  
Vera Kurkova  
Halina Kwasnicka  
Bogdan Kwolek  
Pedro Larranaga  
Inbok Lee  
Kidong Lee  
Jun-Seok Lim  
Hong-Dar Lin  
Rafal Lopatka  
Jacek Mandziuk  
Mariusz Mlynarczuk  
Mariusz Malinowski  
Marcin Mrugalski  
Konrad Markowski  
Francesco Masulli  
Yuri Merkuryev  
Zbigniew Mikrut  
Leila Mokhanche  
Marco Montes de Oca  
Jose Moreno  
Nadia Nedjah  
Roman Neruda  
Mariusz Nieniewski  
Joanna Nowak  
Piotr Nowak  
Marek Ogiela  
Wlodzimierz Ogryczak  
Stanislaw Osowski  
Andrzej Pacut  
Henryk Palus  
Marcin Paprzycki  
Byung Joo Park  
JungYong Park  
Kiejin Park  
Miroslaw Parol  
Krzysztof Patan  
Nikola Pavesic  
David W. Pearson  
Daniel Prusa  
Artur Przelaskowski

Jochen Radmer  
Remigiusz Rak  
Sarunas Raudys  
Kiril Ribarov  
Bernardete Ribeiro  
Martin Rimnac  
Claudio M. Rocco S.  
Miguel Rocha  
Przemyslaw Rokita  
Maciej Romaniuk  
Maciej Slawinski  
Stanislav Saic  
Marcello Sanguineti  
José Santos Reyes  
Henrik Saxen  
Franciszek Seredynski  
Dongmin Shin  
Barbara Siemiatkowska  
Dominik Sierociuk  
Catarina Silva  
Jiri Sima  
Slawomir Skoneczny  
Andrzej Sluzek  
Czeslaw Smutnicki  
Pierre Soille  
Oleksandr Sokolov  
Nigel Steele  
Barbara Strug  
Pawel Strumillo  
Bartlomiej Sulikowski  
Miroslaw Swiercz  
Krzysztof Szczypiorski  
Jarosaw Szostakowski  
Wojciech Szynkiewicz  
Ryszard Tadeusiewicz  
Tatiana Tambouratzis  
Jorge Tavares  
Tomasz Toczyski  
Krzysztof Trojanowski  
George A. Tsihrintzis  
Pavel Vacha  
Armando Vieira  
Wen-Pai Wang  
Slawomir Wierzchon  
Anna Wilbik

Marcin Witczak  
Maciej Wygralak  
Mykhaylo Yatsymirskyy  
Slawomir Zadrozny

Cezary Zielinski  
Stanislaw H. Zak

## **Organizers**

ICANNGA 2007 was organized by the Control Division of the Institute of Control and Industrial Electronics, Faculty of Electrical Engineering, Warsaw University of Technology, Poland.

## Table of Contents – Part II

### Neural Networks

Evolution of Multi-class Single Layer Perceptron . . . . .	1
<i>Sarunas Raudys</i>	
Estimates of Approximation Rates by Gaussian Radial-Basis Functions . . . . .	11
<i>Paul C. Kainen, Věra Kůrková, and Marcello Sanguineti</i>	
Least Mean Square vs. Outer Bounding Ellipsoid Algorithm in Confidence Estimation of the GMDH Neural Networks . . . . .	19
<i>Marcin Mrugalski and Józef Korbicz</i>	
On Feature Extraction Capabilities of Fast Orthogonal Neural Networks . . . . .	27
<i>Bartłomiej Stasiak and Mykhaylo Yatsymirskyy</i>	
Neural Computations by Asymmetric Networks with Nonlinearities . . . . .	37
<i>Naohiro Ishii, Toshinori Deguchi, and Masashi Kawaguchi</i>	
Properties of the Hermite Activation Functions in a Neural Approximation Scheme . . . . .	46
<i>Bartłomiej Beliczynski</i>	
Study of the Influence of Noise in the Values of a Median Associative Memory . . . . .	55
<i>Humberto Sossa, Ricardo Barrón, and Roberto A. Vázquez</i>	
Impact of Learning on the Structural Properties of Neural Networks . . . . .	63
<i>Branko Šter, Ivan Gabrijel, and Andrej Dobnikar</i>	
Learning Using a Self-building Associative Frequent Network . . . . .	71
<i>Jin-Guk Jung, Mohammed Nazim Uddin, and Geun-Sik Jo</i>	
Proposal of a New Conception of an Elastic Neural Network and Its Application to the Solution of a Two-Dimensional Travelling Salesman Problem . . . . .	80
<i>Tomasz Szatkiewicz</i>	
Robust Stability Analysis for Delayed BAM Neural Networks . . . . .	88
<i>Yijing Wang and Zhiqiang Zuo</i>	
A Study into the Improvement of Binary Hopfield Networks for Map Coloring . . . . .	98
<i>Gloria Galán-Marín, Enrique Mérida-Casermeiro, Domingo López-Rodríguez, and Juan M. Ortiz-de-Lazcano-Lobato</i>	

Automatic Diagnosis of the Footprint Pathologies Based on Neural Networks . . . . .	107
<i>Marco Mora, Mary Carmen Jarur, and Daniel Sbarbaro</i>	
Mining Data from a Metallurgical Process by a Novel Neural Network Pruning Method . . . . .	115
<i>Henrik Saxén, Frank Pettersson, and Matias Waller</i>	
Dynamic Ridge Polynomial Neural Networks in Exchange Rates Time Series Forecasting . . . . .	123
<i>Rozaida Ghazali, Abir Jaafar Hussain, Dhiya Al-Jumeily, and Madjid Merabti</i>	
Neural Systems for Short-Term Forecasting of Electric Power Load . . . . .	133
<i>Michał Bąk and Andrzej Bielecki</i>	
Jet Engine Turbine and Compressor Characteristics Approximation by Means of Artificial Neural Networks . . . . .	143
<i>Maciej Lawryńczuk</i>	
Speech Enhancement System Based on Auditory System and Time-Delay Neural Network . . . . .	153
<i>Jae-Seung Choi and Seung-Jin Park</i>	
Recognition of Patterns Without Feature Extraction by GRNN . . . . .	161
<i>Övünç Polat and Tülay Yıldırım</i>	
Real-Time String Filtering of Large Databases Implemented Via a Combination of Artificial Neural Networks . . . . .	169
<i>Tatiana Tambouratzis</i>	
Parallel Realizations of the SAMANN Algorithm . . . . .	179
<i>Sergejus Ivanikovas, Viktor Medvedev, and Gintautas Dzemyda</i>	
A POD-Based Center Selection for RBF Neural Network in Time Series Prediction Problems . . . . .	189
<i>Wenbo Zhang, Xinchen Guo, Chaoyong Wang, and Chunguo Wu</i>	
<b>Support Vector Machines</b>	
Support, Relevance and Spectral Learning for Time Series . . . . .	199
<i>Bernardete Ribeiro</i>	
Support Vector Machine Detection of Peer-to-Peer Traffic in High-Performance Routers with Packet Sampling . . . . .	208
<i>Francisco J. González-Castaño, Pedro S. Rodríguez-Hernández, Rafael P. Martínez-Álvarez, and Andrés Gómez-Tato</i>	
Improving SVM Performance Using a Linear Combination of Kernels . . . . .	218
<i>Laura Dioşan, Mihai Oltean, Alexandrina Rogozan, and Jean-Pierre Pecuchet</i>	
Boosting RVM Classifiers for Large Data Sets . . . . .	228
<i>Catarina Silva, Bernardete Ribeiro, and Andrew H. Sung</i>	

Multi-class Support Vector Machines Based on Arranged Decision Graphs and Particle Swarm Optimization for Model Selection . . . . .	238
<i>Javier Acevedo, Saturnino Maldonado, Philip Siegmann, Sergio Lafuente, and Pedro Gil</i>	
Applying Dynamic Fuzzy Model in Combination with Support Vector Machine to Explore Stock Market Dynamism . . . . .	246
<i>Deng-Yiv Chiu and Ping-Jie Chen</i>	
Predicting Mechanical Properties of Rubber Compounds with Neural Networks and Support Vector Machines . . . . .	254
<i>Mira Trebar and Uroš Lotrič</i>	
An Evolutionary Programming Based SVM Ensemble Model for Corporate Failure Prediction . . . . .	262
<i>Lean Yu, Kin Keung Lai, and Shouyang Wang</i>	

## Biomedical Signal and Image Processing

Novel Multi-layer Non-negative Tensor Factorization with Sparsity Constraints . . . . .	271
<i>Andrzej Cichocki, Rafal Zdunek, Seungjin Choi, Robert Plemmons, and Shun-ichi Amari</i>	
A Real-Time Adaptive Wavelet Transform-Based QRS Complex Detector . . . . .	281
<i>Marek Rudnicki and Paweł Strumiłło</i>	
Nucleus Classification and Recognition of Uterine Cervical Pap-Smears Using FCM Clustering Algorithm . . . . .	290
<i>Kwang-Baek Kim, Sungshin Kim, and Gwang-Ha Kim</i>	
Rib Suppression for Enhancing Frontal Chest Radiographs Using Independent Component Analysis . . . . .	300
<i>Bilal Ahmed, Tahir Rasheed, Mohammed A.U. Khan, Seong Jin Cho, Sungyoung Lee, and Tae-Seong Kim</i>	
A Novel Hand-Based Personal Identification Approach . . . . .	309
<i>Miao Qi, Yinghua Lu, Hongzhi Li, Rujuan Wang, and Jun Kong</i>	
White Blood Cell Automatic Counting System Based on Support Vector Machine . . . . .	318
<i>Tomasz Markiewicz, Stanisław Osowski, and Bożena Mariańska</i>	
Kernels for Chemical Compounds in Biological Screening . . . . .	327
<i>Karol Kozak, Marta Kozak, and Katarzyna Stapor</i>	
A Hybrid Automated Detection System Based on Least Square Support Vector Machine Classifier and $k$ -NN Based Weighted Pre-processing for Diagnosing of Macular Disease . . . . .	338
<i>Kemal Polat, Sadık Kara, Ayşegül Güven, and Salih Güneş</i>	

Analysis of Microscopic Mast Cell Images Based on Network of Synchronised Oscillators . . . . .	346
<i>Michał Strzelecki, Hyongsuk Kim, Paweł Liberski, and Anna Zalewska</i>	
Detection of Gene Expressions in Microarrays by Applying Iteratively Elastic Neural Net . . . . .	355
<i>Máx Chacón, Marcos Lévano, Héctor Allende, and Hans Nowak</i>	
A New Feature Selection Method for Improving the Precision of Diagnosing Abnormal Protein Sequences by Support Vector Machine and Vectorization Method . . . . .	364
<i>Eun-Mi Kim, Jong-Cheol Jeong, Ho-Young Pae, and Bae-Ho Lee</i>	
Epileptic Seizure Prediction Using Lyapunov Exponents and Support Vector Machine . . . . .	373
<i>Bartosz Świdorski, Stanisław Osowski, Andrzej Cichocki, and Andrzej Rysz</i>	
Classification of Pathological and Normal Voice Based on Linear Discriminant Analysis . . . . .	382
<i>Ji-Yeoun Lee, SangBae Jeong, and Minsoo Hahn</i>	
Efficient 1D and 2D Daubechies Wavelet Transforms with Application to Signal Processing . . . . .	391
<i>Piotr Lipinski and Mykhaylo Yatsymirskyy</i>	
A Branch and Bound Algorithm for Matching Protein Structures . . . . .	399
<i>Janez Konc and Dušanka Janežič</i>	

**Biometrics**

Multimodal Hand-Palm Biometrics . . . . .	407
<i>Ryszard S. Choraś and Michał Choraś</i>	
A Study on Iris Feature Watermarking on Face Data . . . . .	415
<i>Kang Ryoung Park, Dae Sik Jeong, Byung Jun Kang, and Eui Chul Lee</i>	
Keystroke Dynamics for Biometrics Identification . . . . .	424
<i>Michał Choraś and Piotr Mroczkowski</i>	
Protecting Secret Keys with Fuzzy Fingerprint Vault Based on a 3D Geometric Hash Table . . . . .	432
<i>Sungju Lee, Daesung Moon, Seunghwan Jung, and Yongwha Chung</i>	
Face Recognition Based on Near-Infrared Light Using Mobile Phone . . . . .	440
<i>Song-yi Han, Hyun-Ae Park, Dal-ho Cho, Kang Ryoung Park, and Sangyoun Lee</i>	
NEU-FACES: A Neural Network-Based Face Image Analysis System . . . . .	449
<i>Ioanna-Ourania Stathopoulou and George A. Tsihrintzis</i>	

GA-Based Iris/Sclera Boundary Detection for Biometric Iris Identification . . . . .	457
<i>Tatiana Tambouratzis and Michael Masouris</i>	
Neural Network Based Recognition by Using Genetic Algorithm for Feature Selection of Enhanced Fingerprints . . . . .	467
<i>Adem Alpaslan Altun and Novruz Allahverdi</i>	
<b>Computer Vision</b>	
Why Automatic Understanding? . . . . .	477
<i>Ryszard Tadeusiewicz and Marek R. Ogiela</i>	
Automatic Target Recognition in SAR Images Based on a SVM Classification Scheme . . . . .	492
<i>Wolfgang Middelman, Alfons Ebert, and Ulrich Thoennesen</i>	
Adaptive Mosaicing: Principle and Application to the Mosaicing of Large Image Data Sets . . . . .	500
<i>Conrad Bielski and Pierre Soille</i>	
Circular Road Signs Recognition with Affine Moment Invariants and the Probabilistic Neural Classifier . . . . .	508
<i>Bogusław Cyganek</i>	
A Context-Driven Bayesian Classification Method for Eye Location . . . .	517
<i>Eun Jin Koh, Mi Young Nam, and Phill Kyu Rhee</i>	
Computer-Aided Vision System for Surface Blemish Detection of LED Chips . . . . .	525
<i>Hong-Dar Lin, Chung-Yu Chung, and Singa Wang Chiu</i>	
Detection of Various Defects in TFT-LCD Polarizing Film . . . . .	534
<i>Sang-Wook Sohn, Dae-Young Lee, Hun Choi, Jae-Won Suh, and Hyeon-Deok Bae</i>	
Dimensionality Problem in the Visualization of Correlation-Based Data . . . . .	544
<i>Gintautas Dzemyda and Olga Kurasova</i>	
A Segmentation Method for Digital Images Based on Cluster Analysis . . . . .	554
<i>Héctor Allende, Carlos Becerra, and Jorge Galbiati</i>	
Active Shape Models and Evolution Strategies to Automatic Face Morphing . . . . .	564
<i>Vittorio Zanella, Héctor Vargas, and Lorna V. Rosas</i>	
Recognition of Shipping Container Identifiers Using ART2-Based Quantization and a Refined RBF Network . . . . .	572
<i>Kwang-Baek Kim, Minhwan Kim, and Young Woon Woo</i>	

A Local-Information-Based Blind Image Restoration Algorithm Using a MLP .....	582
<i>Hui Wang, Nian Cai, Ming Li, and Jie Yang</i>	
Reflective Symmetry Detection Based on Parallel Projection.....	590
<i>Ju-Whan Song and Ou-Bong Gwon</i>	
Detail-Preserving Regularization Based Removal of Impulse Noise from Highly Corrupted Images .....	599
<i>Bogdan Kwolek</i>	
Fast Algorithm for Order Independent Binary Homotopic Thinning ....	606
<i>Marcin Iwanowski and Pierre Soille</i>	
A Perturbation Suppressing Segmentation Technique Based on Adaptive Diffusion .....	616
<i>Wolfgang Middelmann, Alfons Ebert, Tobias Deißler, and Ulrich Thoennessen</i>	
Weighted Order Statistic Filters for Pattern Detection .....	624
<i>Slawomir Skoneczny and Dominik Cieslik</i>	
Real-Time Image Segmentation for Visual Servoing.....	633
<i>Witold Czajewski and Maciej Staniak</i>	

## Control and Robotics

A Neural Framework for Robot Motor Learning Based on Memory Consolidation .....	641
<i>Heni Ben Amor, Shuhei Ikemoto, Takashi Minato, Bernhard Jung, and Hiroshi Ishiguro</i>	
Progressive Optimisation of Organised Colonies of Ants for Robot Navigation: An Inspiration from Nature.....	649
<i>Tatiana Tambouratzis</i>	
An Algorithm for Selecting a Group Leader in Mobile Robots Realized by Mobile Ad Hoc Networks and Object Entropy .....	659
<i>Sang-Chul Kim</i>	
Robot Path Planning in Kernel Space .....	667
<i>José Alí Moreno and Cristina García</i>	
A Path Finding Via VRML and VISION Overlay for Autonomous Robot .....	676
<i>Kil To Chong, Eun-Ho Son, Jong-Ho Park, and Young-Chul Kim</i>	
Neural Network Control for Visual Guidance System of Mobile Robot .....	685
<i>Young-Jae Ryoo</i>	
Cone-Realizations of Discrete-Time Systems with Delays .....	694
<i>Tadeusz Kaczorek</i>	



Global Stability of Neural Networks with Time-Varying Delays . . . . .	704
<i>Yijing Wang and Zhiqiang Zuo</i>	
A Sensorless Initial Rotor Position Sensing Using Neural Network for Direct Torque Controlled Permanent Magnet Synchronous Motor Drive . . . . .	713
<i>Mehmet Zeki Bilgin</i>	
Postural Control of Two-Stage Inverted Pendulum Using Reinforcement Learning and Self-organizing Map . . . . .	722
<i>Jae-kang Lee, Tae-seok Oh, Yun-su Shin, Tae-jun Yoon, and Il-hwan Kim</i>	
Neural Network Mapping of Magnet Based Position Sensing System for Autonomous Robotic Vehicle . . . . .	730
<i>Dae-Yeong Im, Young-Jae Ryoo, Jang-Hyun Park, Hyong-Yeol Yang, and Ju-Sang Lee</i>	
Application of Fuzzy Integral Control for Output Regulation of Asymmetric Half-Bridge DC/DC Converter . . . . .	738
<i>Gyo-Bum Chung</i>	
Obtaining an Optimum PID Controller Via Adaptive Tabu Search . . . . .	747
<i>Deacha Puangdownreong and Sarawut Sujitjorn</i>	
<b>Author Index</b> . . . . .	757

# Evolution of Multi-class Single Layer Perceptron

Sarunas Raudys

Vilnius Gediminas Technical University  
Sauletekio 11, Vilnius, LT-10223, Lithuania  
raudys@ktl.mii.lt

**Abstract.** While training single layer perceptron (SLP) in two-class situation, one may obtain seven types of statistical classifiers including minimum empirical error and support vector (SV) classifiers. Unfortunately, both classifiers cannot be obtained automatically in multi-category case. We suggest designing  $K(K-1)/2$  pair-wise SLPs and combine them in a special way. Experiments using  $K=24$  class chromosome and  $K=10$  class yeast infection data illustrate effectiveness of new multi-class network of the single layer perceptrons.

## 1 Introduction

Among dozens of linear classification algorithms, the SLPs together with SV machines are considered to be among the most effective ones [1], [2], [3], [4], [5] in two pattern class (category) situations. While training two-category nonlinear SLP based classifier in a special way, one may obtain seven different types of classification algorithms [5], [6]. If training sample sizes in two pattern classes  $N_2 = N_1 = N/2$ , a mean vector of training set is moved to a centre of coordinates and we start total gradient training from a weight vector with zero components, then after the first iteration we obtain Euclidean distance classifier (EDC) based on mean vectors of the pattern classes. Afterwards, we move towards linear regularized discriminant analysis, standard linear Fisher classifier or the Fisher classifier with pseudo-inverse of the covariance matrix (for an introduction into statistical pattern recognition see e.g. [2,], [5]). With a progress of iterative adaptation procedure, one has robust discriminant analysis. At the end, when the perceptron weights become large, one may approach the minimum empirical error or maximal margin (support vector) classifiers.

Evolution is a superb peculiarity of total gradient single layer perceptron training procedure enabling us to obtain a sequence of diverse classifiers of increasing complexity. Unfortunately, we cannot profit from this distinctiveness of nonlinear single layer perceptron in multi-category case since: 1) mean vectors of each pair of the pattern classes are different and 2) around decision boundaries between each pair of the pattern classes we have diverse subsets of training vectors. So, after training process terminates, optimal decision boundaries between each pair of pattern classes disagree with that of minimum empirical error or support vector classifiers.

In preceding paper [7], an attention was paid to *starting evolution* of the  $K$ -class SLPs: an improved initialization of the weights was proposed. It was recommended to start training from correctly scaled weights of EDC or regularized Fisher classifier. In order to avoid “training shock” (enormous gradient of the cost function after the

weights initialization), statistically determined initial weight vector was multiplied by iteratively calculated positive constant. Prior to training we suggested performing whitening data transformation on a basis of regularized estimate of covariance matrix supposed to be common for all  $K$  pattern classes. This approach gave a definite gain in small sample size situations. In large sample size situations, more complex decision making rules such as robust, minimum empirical error or support vector classifiers are preferable. Correct initialization of the network becomes less important. So, in present paper, a main attention is focused on the classifiers obtained in *final evolution stage* of the SLPs. Both SLPs and SV classifiers were originally designed for binary classification. Hsu and Lin [8] found that  $K(K-1)/2$  classifications “one-against-one,” outperform a “one-against-all” strategy. Following their recommendations we investigate the one-against-one approach. Instead of SV classifiers we examine optimally stopped SLPs and compare diverse lines of attack to fuse pair-wise decisions.

## 2 Minimum Empirical Error and SV Classifiers in SLP Training

While training SLP with sigmoid activation function in  $K$ -category case, one minimizes a sum of squares cost

$$Cost = \frac{1}{N_1 + N_2 + \dots + N_K} \sum_{l=1}^K \sum_{i=1}^{N_i} [t_{lij} - 1/(1 + \exp(\mathbf{x}_{ij}^T \mathbf{w}_l + w_{l0}))]^2 \quad (1)$$

where  $N_i$  is a number of training vectors of  $i$ -th class,  $\Pi_i$ ,  $t_{lij}$  is desired output,  $\mathbf{w}_l$  is  $p$ -dimensional weight vector and  $w_{l0}$  is a bias term, both corresponding to  $l$ -th category (output), and  $\mathbf{x}_{ij}^T = [x_1, x_2, \dots, x_p]^T$  is transposed  $j$ -th  $p$ -dimensional vector.

In to category case, if targets  $t_j^{(1)} = 1$ ,  $t_j^{(2)} = 0$ , and the weights grow to be very large with a progress of training procedure, outputs,  $1/(1 + (\mathbf{x}_{ij}^T \mathbf{w}_l + w_{l0}))$ , turn out to be close either to 0 or 1. Then cost function (1) starts expressing a fraction of training vectors misclassified: we have a *minimum empirical error classifier*. Let in two category case we have no training errors and the weights are already large. Then *only vectors closest to hyperplane*,  $\mathbf{x}^T \mathbf{w}_{(l)} + w_{0(l)} = 0$ , are contributing to cost (1). E.g., let  $w_0 = 0$ ,  $\mathbf{w}^T = [1 \ 1]$ . Then distances of training vectors  $\mathbf{x}_A^T = [1 \ 1]$  and  $\mathbf{x}_B^T = [1.01 \ 1.01]$  from hyperplane  $\mathbf{x}^T \mathbf{w} + w_0 = 0$  will be  $\sqrt{2} = 1.4142$  and 1.4284, correspondingly. If the vectors A and B are classified correctly, their contribution to the sum in cost function (1) are approximately the same, 0.0142 and 0.0132. If the weights would be 17.5 times larger (i.e.  $w_0 = 0$ ,  $\mathbf{w}^T = [17.7 \ 17.7]$ ), the contribution of vector A would be  $4.5 \times 10^{-31}$  and the contribution of vector's B would be *ten times smaller*. So, vector B will affect decision boundary much more weakly. Consequently, while training the SLP based classifier, we may approach *maximal margin classifier* (SV machine). In multi-category case, however, one ought to remember that around each decision boundary separating any of two pairs of the classes, we have *different subsets of training vectors from each category*. For that reason, while considering the  $K$ -category classification problem from a point of view of

the criteria of maximal margin, the classifier obtained will be not more optimal for each pair of the classes.

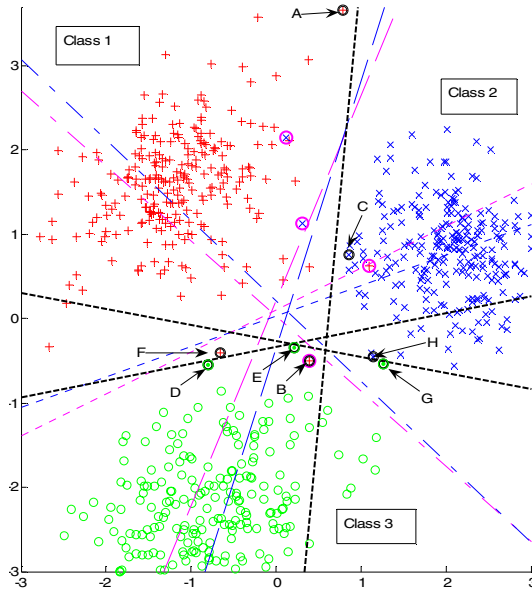
**Two-dimensional example.** To demonstrate particularities arising while training the network of SLPs in multi-category situations we present an example with real world data. While solving  $K=24$  class chromosome classification problem (see Sect. 4) we selected three categories ( $1^{\text{st}}$ ,  $10^{\text{th}}$  and  $11^{\text{th}}$ ), normalized 30-dimensional (30D) input data by moving a mean vector of training data into centre of coordinates and scaling all input variables to have variance of each feature equal to 1. A half of the data, 250 vectors of each category, were used for training. We trained the set of three SLPs at first. After 300 total gradient iterations, empirical classification error stopped to decrease. Outputs of three SLP,  $o_1, o_2, o_3$ , were used to form two-dimensional space, new features  $y_1 = o_1 - o_2, y_2 = o_2 - o_3$ . In Fig. 1, we see 750 2D training vectors of three pattern classes (class  $\Pi_1$  - red pluses,  $\Pi_2$  - blue crosses and  $\Pi_3$  - green circles) in space  $y_1, y_2$ , obtained after shifting data centre, into coordinates centre.

Later on, a set of three SLPs was trained in 2D space starting from initial weights with zero valued components. In this experiment, training was controlled by a test set composed of 750 vectors. Magenta lines show decision boundaries between pairs of the classes  $\Pi_1$ - $\Pi_2$ ,  $\Pi_1$ - $\Pi_3$  and  $\Pi_2$ - $\Pi_3$  after training the network of three SLPs 10000 of iterations ( $P_{\text{validation}} = 0.0227$ ). Blue lines show the boundaries when minimal validation error was obtained ( $P_{\text{validation}} = 0.02$ , after 2200 iterations).

In order to obtain best separation of training sets we trained *three distinct single layer perceptrons* with vectors of  $\Pi_1$ - $\Pi_2$ ,  $\Pi_1$ - $\Pi_3$  or  $\Pi_2$ - $\Pi_3$  categories. One of our aims is to demonstrate that SPL could assist in obtaining *three different SV classifiers* to discriminate the pairs of the pattern classes. The SV classifier could be obtained when the perceptron weights are large enough. Then only the vectors closed to decision boundary contribute to the cost function and determine the perceptron weights.

When the weights are large, the gradient is small and training becomes slow. In order to ensure training process, we increased learning step gradually: after each iteration, the learning step was increased by factor  $\gamma > 1$ . For pair of the classes  $\Pi_2$ - $\Pi_3$  we succeeded obtaining errorless classification. For this pair we fixed  $\gamma = 1.001$  and trained the perceptron 13,000 of iterations. Then learning step,  $\eta_{\text{final}}$ , became 1301.3. Further increase in the number of iterations caused a divergence of total gradient training algorithm. For pairs  $\Pi_1$ - $\Pi_2$  and  $\Pi_1$ - $\Pi_3$  we have *non-zero empirical classification error*. Therefore, we were more prudent and had chosen smaller factor  $\gamma$  ( $\gamma = 1.0001$ ). Training process was performed almost up to divergence: 50,000 of iterations for pair  $\Pi_1$ - $\Pi_2$  (then,  $\eta_{\text{final}} = 5000.5$ ) and 805,000 iterations for pair  $\Pi_1$ - $\Pi_3$  ( $\eta_{\text{final}} = 80508$ ). We see *the control of learning step is vital* in obtaining SV classifier.

Support vectors A, B of class  $\Pi_1$  and vector C of class  $\Pi_2$  determine SV decision boundary  $SV_{12}$  between classes  $\Pi_1$  and  $\Pi_2$ . Support vectors D, E of class  $\Pi_3$  (green circles) and vector F of class  $\Pi_1$  (red plus) determine SV decision boundary  $SV_{13}$  between classes  $\Pi_1$  and  $\Pi_3$ . Support vectors E, G of class  $\Pi_3$  and vector H of class  $\Pi_2$  determine SV decision boundary between classes'  $\Pi_2$  and  $\Pi_3$ . Two vectors of class  $\Pi_1$  and one vector of  $\Pi_2$  (all three marked by bold magenta circles) are *misclassified* by decision boundary  $SV_{12}$ , however, *do not affect the position of this boundary*. The same could be said about vector B (also marked by bold magenta circle) which is misclassified by  $SV_{13}$ .



**Fig. 1.** Demonstration that near decision boundaries of any of two pairs of pattern classes we have different subsets of training vectors from each of the classes. Training set composed of 250 vectors of each class (class  $\Pi_1$  red pluses,  $\Pi_2$  – blue crosses and  $\Pi_3$  – green circles), decision boundaries of three class net of SLPs (magenta and blue) and decision boundaries of three pair-wise SV classifiers (bold black dotted lines).

“Soft margins” is a positive feature of non-linear SLP prudently trained by total gradient. The decision boundaries of the net of SLPs meet at unique points. The pair-wise decision boundaries, however, intersect at diverse points and form a region where classification is ambiguous. In Fig. 1 we see a small triangle where all three pair-wise decision boundaries allocate vectors to three different classes. What to do? If the region of the ambiguous decision is small, we need to have some other high-quality decision making rule which would make final decision.

### 3 Practical Aspects of Training the $K$ -Class Network of SLPs

**Training pair-wise single layer perceptrons.** In order to profit from evolution of the SLP during development of training procedure, we suggest performing classification by means of  $K(K-1)/2$  pair-wise linear discriminant functions generated after training of the perceptron and fuse the pair-wise decisions in order to allocate unknown vector  $x$  to one of the pattern classes. To solve this task perfectly, for each pair of the classes we have to train individual perceptron in the best possible way, i.e. we train it in a special manner and stop training on a right moment.

The main two requirements to profit from evolution of non-linear SLP classifier is to move training data mean,  $\hat{\mu}_{ij}$ , of the pair of classes,  $\Pi_i$  and  $\Pi_j$ , into the centre of

coordinates, and start training from the weight vector with zero components. If a ratio of training set size to dimensionality is small, the covariance matrix of the data could become nearly singular. Eigen-values of the covariance matrix could differ in billions of times. For that reason, training of the perceptron becomes very slow. To speed up training process and to reduce the generalization error for each pair of the classes it is worth to perform whitening data transformation [5], [9], [10]

$$\mathbf{y} = \mathbf{G}_{ij} (\mathbf{x} - \hat{\boldsymbol{\mu}}_{ij}), \quad (2)$$

where  $\mathbf{G}_{ij} = \boldsymbol{\Lambda}^{-1/2} \boldsymbol{\Phi}^T$ , and  $\boldsymbol{\Lambda}$ ,  $\boldsymbol{\Phi}^T$  are eigen-values and eigenvectors of pooled sample covariance matrix,  $\mathbf{S} = \sum N_i \mathbf{S}_i / \sum N_i$ , and  $\mathbf{S}_i$  is sample covariance matrices of class,  $\Pi_i$ .

In small training sample situations, one can make use of a variety of statistical methods to improve sample estimate of covariance matrix. The simplest way is to use regularized estimate of the matrix [2], [5],  $\mathbf{S}_{\text{regularized}} = \mathbf{S} + \lambda \mathbf{I}$ , where  $\mathbf{I}$  stands for  $p \times p$  identity matrix, and  $\lambda$  is a regularization constant to be selected in an experimental way. If certain prior information about a structure of the covariance matrix exists, it could be utilized by constraining the sample estimate of covariance matrix. As a result, the matrix is determined by smaller number of parameters [11], [12]. In this way, we improve sample size / dimensionality ratio (for more details about integration of statistical methods into SLP training see Chap. 5 in [5]).

Important phase in SLP training is *determination of optimal number of iterations*. While training, the classification algorithms move gradually from simplest statistical methods to more sophisticated ones. Therefore, *determination of stopping moment in fact is determination of optimal complexity of the classifier*. Very often conditional distributions densities of the input pattern vectors are very complex and cannot be approximated by some theoretical model. For that reason, instead of using exact analytical formulae or error bounds, one is obliged to use validation set in order to determine correct stopping moment. The designer has no problems if she/he may take apart a portion of design set vectors in order to form validation set. If the design set is small, the designer faces a problem which proportion of design data to use as training set and which proportion one is obliged to allocate to validation set.

**Use of a noise injection in order to determine stopping moment.** One of possible solutions is to use all design set vectors as training set and form validation set from training vectors by means of a noise injection. A noise injection actually introduces *additional information that declares that a space between nearest vectors of a single pattern class is not empty, but instead is filled up with vectors of the same category*. Colored  $k$ -NN noise injection was suggested to reduce data distortion [13]. To generate such noise, for each single training vector,  $\mathbf{x}_{sl}$ , one finds its  $k$  nearest neighbors of the same pattern class and adds a noise only in a subspace formed by vector  $\mathbf{x}_{sl}$  and  $k$  neighboring training vectors,  $\mathbf{x}_{sl1}$ ,  $\mathbf{x}_{sl2}$ , ...,  $\mathbf{x}_{slk}$ . Random Gaussian,  $N(0, \sigma_{\text{noise}}^2)$ , noise is added  $ni_{nn}$  times along  $k$  lines connecting  $\mathbf{x}_{sl}$  and  $\mathbf{x}_{sl1}$ ,  $\mathbf{x}_{sl2}$ , ...,  $\mathbf{x}_{slk}$ . Three parameters are required to realize the noise injection procedure: 1)  $k$ , the number of neighbors, 2)  $ni_{nn}$ , the number of new, artificial vectors generated around each single training vector,  $\mathbf{x}_{sl}$ , and 3)  $\sigma_{\text{noise}}$ , the noise standard deviation. The noise variance,  $\sigma_{\text{noise}}^2$ , has to be selected as a trade-off between the complexity of the decision boundary and the

learning set size. When working with unknown data, one has to test several values of  $\sigma_{\text{noise}}$  and select the most suitable one [14], [15]. To speed up calculations in our experiments, we used “default” values:  $k=2$ ;  $\sigma_{\text{noise}}=1.0$ ,  $n_{\text{nn}}=25$ .

**Making final decision.** After obtaining  $K(K-1)/2$  pair-wise classifications made by  $K(K-1)/2$  single layer perceptrons, for each unknown vector,  $\mathbf{x}$ , we need to make a final categorization. Popular methods to combine outputs of pair-wise classifications are voting and a method suggested by Hastie and Tibshirani [8], [16], [17]. Blow we suggest using two new alternative fusion methods. In both of them, we allocate vector  $\mathbf{x}$  to class  $\Pi_z$ , if  $K-1$  pair-wise classifiers out of  $K(K-1)/2$  ones are allocating this vector to single pattern class,  $\Pi_z$ . Otherwise, we perform *a second stage of decision making*. In first fusion algorithm, we perform final categorization by the  $K$ -class net of SLPs. Here, both the pair-wise decisions and the fusion are performed by the hyperplanes. It could become a weakness of the method. In order to increase diversity of decision making procedure in initial and final classifications, in the second version of decision making algorithm, final allocation of vector  $\mathbf{x}$  is performed by local classification rule, the kernel discriminant analysis (KDA). The latter procedure is similar to fusion rule of Giacinto *et al.* [18] suggested in Multiple classifier systems (MCS) framework (for recent review of fusion rules see e.g. [19]).

The  $K$ -class set of SLPs was already considered above. In the KDA we perform classification according to nonparametric local estimates of conditional probability density functions of input vectors,  $f_{\text{KDA}}(\mathbf{x} | \Pi_i)$ , and  $q_i$ , prior probabilities of the classes  $i = 1, 2, \dots, K$ . In the experiments reported below, we used Gaussian kernel and performed classification according to the maximum of products

$$q_i \times f_{\text{KDA}}(\mathbf{x} | \Pi_i) = \frac{q_i}{N_i} \sum_{j=1}^{N_i} \exp(-h^{-1}(\mathbf{x} - \mathbf{x}_{ij})^T \mathbf{S}_{Ri}^{-1}(\mathbf{x} - \mathbf{x}_{ij})), \quad (i = 1, 2, \dots, K), \quad (3)$$

where  $h$  is a smoothing parameter,  $\mathbf{S}_{Ri}$  is sample estimate of the  $i^{\text{th}}$  class covariance matrix. Index “R” symbolizes that the matrix estimate could be constrained or regularized to go with small learning set size requirements. To have fair comparison of the KDA based algorithm with other ones, we used default value,  $h = 1.0$ .

## 4 Real World Pattern Recognition Tasks

**Data.** In order to look into usefulness of new strategy to design the network of single layer perceptrons to solve multi-class pattern recognition problems, two hot real world biomedical problems were investigated. The first data set were the chromosome 30-dimensional band profiles [20], consisting of 24 classes, 500 vectors in each of them. The second problem was 1500-dimensional spectral yeast infection data. The identification and analysis of closely related species of yeast infections is problematic [21]. For analysis we had 1001 data vectors from ten different infections ( $K=10$ ): 113, 84, 116, 83, 120, 56, 90, 97, 113 and 129 vectors in each of them.

**Experiments.** In evaluation of different variants of learning procedures we used two-fold cross validation technique. Every second vector of each pattern class was selected for training. Remaining vectors were used for testing of the classification algorithms.

In order to obtain more reliable results, the cross validation procedures were repeated 25 times after preceding reshuffling of the data vectors in each category.

Focus of present paper is *final evolution* of the network of the pair-wise perceptrons in the situation where in final stage of learning process the minimum empirical error or support vector classifiers are obtained. One may argue that these classification algorithms are most complex in the sequence of methods that could be obtained while training nonlinear SLP.

Dimensionality  $p=1500$  for 50 vectors from each category allocated for training in the experiments with yeast data, is too high to obtain reliable SV or minimum empirical error classifiers. So, for dimensionality reduction we used slightly regularized ( $\lambda = 0.01$ )  $K$ -class Fisher classifier. Ten outputs of the Fisher classifiers were mapped into 9D space. Next, a mean of the 9D training set was moved into a centre of coordinates and all features were scaled to equalize standard deviations.

To determine optimal number of training epochs of  $K$  class SLP (KSLP) and two variants of pair-wise SLPs (pwSLP+KSLP and pwSLP+KDA) we used artificially generated pseudo-validation set. This set was generated by means of colored noise injection as described in Sect. 3. Thus, in the experiments with yeast data we used  $\sim 12,500$  9D vectors for validation. We used 150,000 30D artificially generated validation vectors in the experiments with chromosome data.

**Results.** Average values of generalization errors evaluated in  $2 \times 25 = 50$  cross-validation runs of the experiments with randomly permuted data are presented in Table 1. In column “KSLP/amb/PW” we present: 1) average generalization error of  $K$  class net of SLPs, 2) a fraction of *ambiguous* vectors where  $K-1$  perceptrons allocated the test vectors to more than to one pattern class, 3) average generalization error of KDA. In subsequent four columns we present averages of generalization errors where pair-wise decisions of  $K(K-1)/2$  single layer perceptrons (on the left side of each column) or SV classifiers (on the right side) were fused by four different techniques: Voting, Hastie-Tibshirani method,  $K$  class net of SLP or KDA as described above. To obtain the pair-wise SV classifiers we used Chang and Lin software [22]. Standard deviations of the averages were  $\sim 0.2\%$  (for chromosome data) and  $\sim 0.07\%$  (for yeast data). The *best strategies* are printed in **bold**.

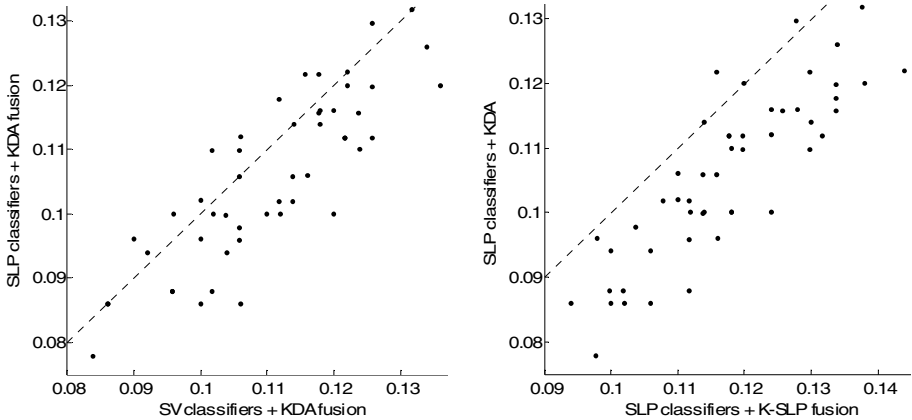
Experimental investigations advocate that on average all four combinations of pair-wise classifications represented as MCSs outperform single stage classification algorithms, the  $K$  class net of SLPs and Kernel discriminant analysis. In all four pair-wise decision fusion methods, optimally stopped single layer perceptrons as adaptive and more flexible method, outperformed MCS composed of maximal margin (support vector) classifiers. In all experiments, local fusion (KDA) of the pair-wise decisions outperformed the global fusion algorithms realized by the hyperplanes ( $K$  class net of SLPs, majority voting or H-T method).

**Table 1.** Average generalization errors (in %) in 50 experimental runs with real world data

Data set, dimensionality	KSLP/amb/KDA	+ Voting	+ H-T	+ KSLP	+ KDA
Yeast, $K=10, p=1500 \rightarrow 9$	13.5 / $1/_{12}$ / 13.1	13.3/15.9	13.3/15.0	11.7/11.4	<b>10.6/11.1</b>
Chromosomes, $K=24, p=30$	26.6 / $1/_{4}$ / 26.9	19.8/21.9	19.6/21.4	17.4/18.4	<b>14.2/15.9</b>



In all 50 experiments with chromosome data, pair-wise SLP with KDA outperformed other methods. In experiments with yeast data, this method was the best on average: in certain experiments out of 50, it was outperformed by other methods (see Fig. 2).



**Fig. 2.** Distributions of generalization errors in the 50 experiments with yeast data:  $K(K-1)/2$  pair-wise SLPs with KDA as fusion rule (y axis) versus  $K(K-1)/2$  pair-wise SV classifiers with the KDA as fusion rule (on the left) and pair-wise SLPs with fusion rule formed of  $K$  class net of single layer perceptrons (on the right)

We stress once more that all algorithms were compared in identical conditions: the artificially generated validation sets were used to determine stopping moments, single *a priori* fixed smoothing parameter value ( $h=1$ ) was used in KDA.

## 5 Concluding Remarks

Previous theoretical analysis has shown that, in principle, one may obtain seven different types of the classification algorithms while training two-category non-linear single layer perceptron by total gradient algorithm in a special way. At the beginning of training procedure, we may obtain a sequence of four statistical classifiers based on model of multivariate Gaussian density with common covariance matrix. After that, normality assumptions are “released” gradually: we move closer to robust linear classifier that ignores training vectors distant from decision hyperplane. At the end, we may approach the minimum empirical error and support vector classifiers which take into account only training vectors closest to decision hyperplane. The latter fact becomes very important in the multi-category case, where at closest neighborhood of the hyperplanes discriminating each pair of the pattern classes, we have diverse subsets of training vectors. For that reason, we cannot profit from this magnificent peculiarity of evolution of nonlinear single layer perceptron: decision boundaries between each pair of pattern classes differ from that of minimum empirical error and support vector classifiers.

In order to profit from evolution of the single layer perceptron during development of training procedure, like in MCSs we suggest performing decision making in two stages. At first, we classify unknown vector,  $\mathbf{x}$ , by means of  $K(K-1)/2$  single layer perceptrons optimally stopped for each pair of the pattern classes. Thus, for each pair of the classes we have the classifier of optimal complexity. We assign unknown vector to  $i$ -th class if  $K-1$  pair-wise discriminant functions are classifying this vector to single class,  $\Pi_i$ . If the first stage classifiers disagree, for final allocation of vector  $\mathbf{x}$  we suggest using local classification method, the kernel discriminant analysis.

The experiments performed with two real world data sets and multiple splits of data into training and test sets do not generalize, however, supported by strong theoretical considerations about optimality of pair-wise decisions and diversity of base (pair-wise) classifiers and the KDA fusion rule give heavy arguments that two stage decision making strategy described above is promising and worth practical application and further theoretical investigation. In future we suggest investigating of more flexible fusion rules. Such rules could be: a) the KDA with adaptive smoothing parameter,  $h$ , b) radial basis function neural networks, c) reduction of the number of categories in final decision makings. No doubt, the experiments with larger number of real world data sets should be conducted. Possibly, the main problem, however, should be a complexity analysis of the pair-wise classifiers and the fusion rules in relation to input dimensionality and the design set size [15].

**Acknowledgements.** The author is thankful to Prof. R.P.W. Duin from Delft University of Technology and Prof. R. Somorjai from Institute of Biodiagnostics, Winnipeg for useful discussions and biomedical data provided for the experiments.

## References

1. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge Univ. Press, Cambridge, UK (2000)
2. Duda, R.O., Hart P.E., Stork D.G.: Pattern Classification. 2nd edn. Wiley, NY (2000).
3. Haykin, S.: Neural Networks: A comprehensive foundation. 2nd edn. Prentice-Hall, Englewood Cliffs, NJ (1999)
4. Raudys, S.: How good are support vector machines? Neural Networks 13 (2000) 9-11
5. Raudys, S.: Statistical and Neural Classifiers: An integrated approach to design. Springer-Verlag, London Berlin Heidelberg (2001)
6. Raudys, S.: Evolution and generalization of a single neurone. I. SLP as seven statistical classifiers. Neural Networks, 11 (1998) 283–296
7. Raudys S., Denisov V., Bielskis A.: A pool of classifiers by SLP: A multi-class case. Lecture Notes in Computer Science, Vol. 4142 Springer-Verlag, Berlin Heidelberg New York (2006) 47 – 56.
8. Hsu, C. W., Lin C. J.: A comparison on methods for multi-class support vector machines. IEEE Trans. on Neural Networks, 13 (2002) 415-425
9. Le Cun Y., Kanter I., Solla, S.: Eigenvalues of covariance matrices: application to neural-network learning. Physical Review Letters 66 (1991) 2396–2399
10. Halkaer S., Winter O.: The effect of correlated input data on the dynamics of learning. In: Mozer, M.C., Jordan, M.I., Petsche T. (eds.): Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA. 9 (1996) 169–75

11. Saudargiene, A.: Structurization of the covariance matrix by process type and block diagonal models in the classifier design. *Informatica* 10 (1999) 245–269
12. Raudys, S., Saudargiene, A.: First-order tree-type dependence between variables and classification performance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. PAMI-23 (2001) 233-239
13. Duin R.P.W.: Nearest neighbor interpolation for error estimation and classifier optimization. In: Hogd, K.A, Braathen, B., Heia, K. (eds.): *Proc. of the 8th Scandinavian Conference on Image Analysis*, Tromso, Norway (1993) 5-6.
14. Skurichina, M., Raudys, S., Duin R.P.W.: K-NN directed noise injection in multilayer perceptron training, *IEEE Trans. on Neural Networks*, 11 (2000) 504–511
15. Raudys, S.: Trainable Fusion Rules. II. Small sample-size effects. *Neural Networks* 19 (2006) 1517-1527
16. Hastie, T., Tibshirani, R.: Classification by pair-wise coupling. *The Annals of Statistics* 26 (1998) 451-471
17. Wu, T.-F, Lin C.-J., Weng, R.C.: Probability estimates for multi-class classification by pair-wise coupling. *J. of Machine Learning Research* 5 (2004) 975-1005
18. Giacinto, G, Roli F., Fumera G.: Selection of classifiers based on multiple classifier behaviour. *Lecture Notes in Computer Science*, Vol. 1876. Springer-Verlag, Berlin Heidelberg New York (2000) 87–93
19. Raudys, S.: Trainable Fusion Rules. I. Large sample size case. *Neural Networks*, 19 (2006) 1506-1516
20. Pekalska, E., Duin R.P.W.: Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters* 23 (2002) 943–956
21. Pizzi, N.J., Pedrycz, W.: Classification of magnetic resonance spectra using parallel randomized feature selection. *Proc. IJCNN04* (2004).
22. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines Available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm> (2001)

# Estimates of Approximation Rates by Gaussian Radial-Basis Functions

Paul C. Kainen<sup>1</sup>, Věra Kůrková<sup>2</sup>, and Marcello Sanguineti<sup>3</sup>

<sup>1</sup> Department of Mathematics, Georgetown University  
Washington, D. C. 20057-1233, USA  
kainen@georgetown.edu

<sup>2</sup> Institute of Computer Science, Academy of Sciences of the Czech Republic  
Pod Vodárenskou věží 2, Prague 8, Czech Republic  
vera@cs.cas.cz

<sup>3</sup> Department of Communications, Computer, and System Sciences (DIST)  
University of Genoa, Via Opera Pia 13, 16145 Genova, Italy  
marcello@dist.unige.it

**Abstract.** Rates of approximation by networks with Gaussian RBFs with varying widths are investigated. For certain smooth functions, upper bounds are derived in terms of a Sobolev-equivalent norm. Coefficients involved are exponentially decreasing in the dimension. The estimates are proven using Bessel potentials as auxiliary approximating functions.

## 1 Introduction

Gaussian radial-basis functions (RBF) are known to be able to approximate with an arbitrary accuracy all continuous and all  $\mathcal{L}^2$ -functions on compact subsets of  $\mathbb{R}^d$  (see, e.g., [7], [16], [17], [18], [19]). In such approximations, the number of RBF units plays the role of a measure of model complexity, which determines the feasibility of network implementation. For Gaussian RBFs with a fixed width, rates of decrease of approximation error with increasing number of units were investigated in [5], [6], [11], [10].

In this paper, we investigate rates of approximation by Gaussian RBF networks with varying widths. As the set of linear combinations of scaled Gaussians is the same as the set of linear combinations of their Fourier transforms, by Plancherel's equality the  $\mathcal{L}^2$ -errors in approximation of a function and its Fourier transform are the same. We exploit the possibility of alternating between function and Fourier transform to obtain upper bounds on rates of approximation. Our bounds are obtained by composing two approximations. As auxiliary approximating functions we use certain special functions called Bessel potentials. These potentials characterize functions belonging to Sobolev spaces.

The paper is organized as follows. In Sect. 2, some concepts, notations and auxiliary results for investigation of approximation by Gaussian RBF networks are introduced. Using an integral representation of the Bessel potential and its Fourier transform in terms of scaled Gaussians, in Sect. 3 we derive upper bounds on rates of approximation of Bessel potentials by linear combinations of scaled

Gaussians. In Sect. 4 where Bessel potentials are used as approximators, upper bounds are derived for functions from certain subsets of Sobolev spaces. In Sect. 5, estimates from the previous two sections are combined to obtain a bound for approximation of certain smooth functions by Gaussian RBFs. Due to space limitations, we have excluded proofs; they are given in 9.

## 2 Approximation by Gaussian RBF Networks

In this paper, we consider approximation error measured by  $\mathcal{L}^2$ -norm with respect to Lebesgue measure  $\lambda$ . Let  $\mathcal{L}^2(\Omega)$  and  $\mathcal{L}^1(\Omega)$  denote, respectively, the space of square-integrable and absolutely integrable functions on  $\Omega \subseteq \mathbb{R}^d$  and  $\|\cdot\|_{\mathcal{L}^2(\Omega)}$ ,  $\|\cdot\|_{\mathcal{L}^1(\Omega)}$  the corresponding norms. When  $\Omega = \mathbb{R}^d$ , we write merely  $\mathcal{L}^2$  and  $\mathcal{L}^1$ .

For nonzero  $f$  in  $\mathcal{L}^2$ ,  $f^\circ = f/\|f\|_{\mathcal{L}^2}$  is the *normalization* of  $f$ . For  $F \subset \mathcal{L}^2$ ,  $F|_\Omega$  denotes the set of functions from  $F$  restricted to  $\Omega$ ,  $\hat{F}$  is the set of Fourier transforms of functions in  $F$ , and  $F^\circ$  the set of normalizations. For  $n \geq 1$ , we define  $\text{span}_n F := \{\sum_{i=1}^n w_i f_i \mid f_i \in F, w_i \in \mathbb{R}\}$ .

In a normed linear space  $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$ , for  $f \in \mathcal{X}$  and  $A \subset \mathcal{X}$ , we shall write  $\|f - A\|_{\mathcal{X}} = \inf_{g \in A} \|f - g\|_{\mathcal{X}}$ . Two norms  $\|\cdot\|$  and  $|\cdot|$  on the same linear space are *equivalent* if there exists  $\kappa > 0$  such that  $\kappa^{-1}\|\cdot\| \leq |\cdot| \leq \kappa\|\cdot\|$ .

A *Gaussian radial-basis-function unit* with  $d$  inputs computes all scaled and translated Gaussian functions on  $\mathbb{R}^d$ . For  $b > 0$ , let  $\gamma_b : \mathbb{R}^d \rightarrow \mathbb{R}$  be the scaled Gaussian defined by

$$\gamma_b(x) = \exp(-b\|x\|^2) = e^{-b\|x\|^2}.$$

Then a simple calculation shows that

$$\|\gamma_b\|_{\mathcal{L}^2} = (\pi/2b)^{d/4}. \quad (1)$$

Let

$$G = \{\tau_y \gamma_b \mid y \in \mathbb{R}^d, b > 0\} \text{ where } (\tau_y f)(x) = f(x - y)$$

denote the set of translations of scaled Gaussians and

$$G_0 = \{\gamma_b \mid b \in \mathbb{R}_+\}$$

the set of scaled Gaussians centered at 0.

We investigate rates of approximation by *networks with  $n$  Gaussian RBF units and one linear output unit*, which compute functions from the set  $\text{span}_n G$ .

The  *$d$ -dimensional Fourier transform* is the operator  $\mathcal{F}$  on  $\mathcal{L}^2 \cap \mathcal{L}^1$  given by

$$\mathcal{F}(f)(s) = \hat{f}(s) = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{ix \cdot s} f(x) dx, \quad (2)$$

where  $\cdot$  denotes the Euclidean inner product on  $\mathbb{R}^d$ . The Fourier transform behaves nicely on  $G_0$ : for every  $b > 0$ ,

$$\widehat{\gamma_b}(x) = (2b)^{-d/2} \gamma_{1/4b}(x) \quad (3)$$

(cf. [22, p. 43]). Thus

$$\text{span}_n G_0 = \text{span}_n \widehat{G}_0. \quad (4)$$

Calculation shows that

$$\|\widehat{\gamma}_b\|_{\mathcal{L}^2} = (2b)^{-d/2} (2b\pi)^{d/4} = \|\gamma_b\|_{\mathcal{L}^2} \quad (5)$$

and Plancherel's identity [22, p. 31] asserts that Fourier transform is an isometry on  $\mathcal{L}^2$ : for all  $f \in \mathcal{L}^2$

$$\|f\|_{\mathcal{L}^2} = \|\widehat{f}\|_{\mathcal{L}^2}. \quad (6)$$

By (6) and (4) we get the following equality.

**Proposition 1.** *For all positive integers  $d, n$  and all  $f \in \mathcal{L}^2$ ,*  
 $\|f - \text{span}_n G_0\|_{\mathcal{L}^2} = \|f - \text{span}_n \widehat{G}_0\|_{\mathcal{L}^2} = \|\widehat{f} - \text{span}_n \widehat{G}_0\|_{\mathcal{L}^2} = \|\widehat{f} - \text{span}_n G_0\|_{\mathcal{L}^2}.$

So in estimating rates of approximation by linear combinations of scaled Gaussians, one can switch between a function and its Fourier transform.

To derive our estimates, we use a result on approximation by convex combinations of  $n$  elements of a bounded subset of a Hilbert space derived by Maurey [20], Jones [8] and Barron [2,3]. Let  $F$  be a bounded subset of a Hilbert space  $(\mathcal{H}, \|\cdot\|_{\mathcal{H}})$ ,  $s_F = \sup_{f \in F} \|f\|_{\mathcal{H}}$ , and  $\text{uconv}_n F = \{\sum_{i=1}^n \frac{1}{n} f_i \mid f_i \in F\}$  denote the set of  $n$ -fold convex combinations of elements of  $F$  with all coefficients equal. By Maurey-Jones-Barron's result [3, p. 934], for every function  $h$  in  $\text{cl conv}(F \cup -F)$ , i.e., from the closure of the symmetric convex hull of  $F$ , we have

$$\|h - \text{uconv}_n F\|_{\mathcal{H}} \leq \frac{s_F}{\sqrt{n}}. \quad (7)$$

The bound (7) implies an estimate of the distance from  $\text{span}_n F$  holding for any function from  $\mathcal{H}$ . The estimate is formulated in terms of a norm tailored to  $F$ , called  $F$ -variation and denoted by  $\|\cdot\|_F$ . It is defined for any bounded subset  $F$  of any normed linear space  $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$  as the Minkowski functional of the closed convex symmetric hull of  $F$  (closure with respect to the norm  $\|\cdot\|_{\mathcal{X}}$ ), i.e.,

$$\|h\|_F = \inf \{c > 0 \mid c^{-1}h \in \text{cl conv}(F \cup -F)\}. \quad (8)$$

Note that  $F$ -variation can be infinite (when the set on the right-hand side is empty) and that it depends on the ambient space norm. In the next sections, we only consider variation with respect to the  $\mathcal{L}^2$ -norm. The concept of  $F$ -variation was introduced in [12] as an extension of "variation with respect to half-spaces" defined in [2], see also [13].

It follows from (7) that for all  $h \in \mathcal{H}$  and all positive integers  $n$ ,

$$\|h - \text{span}_n F\|_{\mathcal{H}} \leq \frac{\|h\|_{F^\circ}}{\sqrt{n}}. \quad (9)$$

It is easy to see that if  $\psi$  is any linear isometry of  $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$ , then for any  $f \in \mathcal{X}$ ,  $\|f\|_F = \|\psi(f)\|_{\psi(F)}$ . In particular,

$$\|\widehat{f}\|_{G_0^\circ} = \|f\|_{G_0^\circ} \quad (10)$$

To combine estimates of variations with respect to two sets, we use the following lemma. Its proof follows easily from the definition of variation.

**Lemma 1.** *Let  $F, H$  be nonempty, nonzero subsets of a normed linear space  $X$ . Then for every  $f \in X$ ,  $\|f\|_F \leq (\sup_{h \in H} \|h\|_F) \|f\|_H$ .*

For  $\phi : \Omega \times Y \rightarrow \mathbb{R}$ , let  $F_\phi = \{\phi(\cdot, y) : \Omega \rightarrow \mathbb{R} \mid y \in Y\}$ . The next theorem is an extension of [14, Theorem 2.2]. Functions on subsets of Euclidean spaces are *continuous almost everywhere* if they are continuous except on a set of Lebesgue measure zero.

**Theorem 1.** *Let  $\Omega \subseteq \mathbb{R}^d$ ,  $Y \subseteq \mathbb{R}^p$ ,  $w : Y \rightarrow \mathbb{R}$ ,  $\phi : \Omega \times Y \rightarrow \mathbb{R}$  be continuous almost everywhere such that for all  $y \in Y$ ,  $\|\phi(\cdot, y)\|_{\mathcal{L}^2(\Omega)} = 1$ , and  $f \in \mathcal{L}^1(\Omega) \cap \mathcal{L}^2(\Omega)$  be such that for all  $x \in \Omega$ ,  $f(x) = \int_Y w(y) \phi(x, y) dy$ . Then  $\|f\|_{F_\phi} \leq \|w\|_{\mathcal{L}^1(Y)}$ .*

### 3 Approximation of Bessel Potentials by Gaussian RBFs

In this section, we estimate rates of approximation by  $\text{span}_n G$  for certain special functions, called Bessel potentials, which are defined by means of their Fourier transforms. For  $r > 0$ , the *Bessel potential* of order  $r$ , denoted by  $\beta_r$ , is the function on  $\mathbb{R}^d$  with Fourier transform

$$\hat{\beta}_r(s) = (1 + \|s\|^2)^{-r/2}.$$

To estimate  $G_0^0$ -variations of  $\beta_r$  and  $\hat{\beta}_r$ , we use Theorem 1 with representations of these two functions as integrals of scaled Gaussians.

For  $r > 0$ , it is known ([21, p. 132]) that  $\beta_r$  is non-negative, radial, exponentially decreasing at infinity, analytic except at the origin, and a member of  $\mathcal{L}^1$ . It can be expressed as an integral

$$\beta_r(x) = c_1(r, d) \int_0^\infty \exp(-t/4\pi) t^{-d/2+r/2-1} \exp(-(\pi/t)\|x\|^2) dt, \quad (11)$$

where  $c_1(r, d) = (2\pi)^{d/2} (4\pi)^{-r/2} / \Gamma(r/2)$  (see [15, p. 296] or [21]). The factor  $(2\pi)^{d/2}$  occurs since we use Fourier transform (2) which includes the factor  $(2\pi)^{-d/2}$ . Rearranging ([1]) slightly, we get a representation of the Bessel potential as an integral of normalized scaled Gaussians.

**Proposition 2.** *For every  $r > 0$ ,  $d$  a positive integer, and  $x \in \mathbb{R}^d$*

$$\beta_r(x) = \int_0^{+\infty} v_r(t) \gamma_{\pi/t}^o(x) dt,$$

where  $v_r(t) = c_1(r, d) \|\gamma_{\pi/t}\|_{\mathcal{L}^2} \exp(-t/4\pi) t^{-d/2+r/2-1} \geq 0$ .

Let  $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$  be the Gamma function (defined for  $z > 0$ ) and put

$$k(r, d) := \frac{(\pi/2)^{d/4} \Gamma(r/2 - d/4)}{\Gamma(r/2)}.$$

By calculating the  $\mathcal{L}^1$ -norm of the weighting function  $v_r$  and applying Theorem [1](#), we get an estimate of  $G_0^\circ$ -variation of  $\beta_r$ .

**Proposition 3.** *For  $d$  a positive integer and  $r > d/2$ ,*

$$\int_0^\infty v_r(t) dt = k(r, d) \quad \text{and} \quad \|\beta_r\|_{G^\circ} \leq \|\beta_r\|_{G_0^\circ} \leq k(r, d).$$

Also the Fourier transform of the Bessel potential can be expressed as an integral of normalized scaled Gaussians.

**Proposition 4.** *For every  $r > 0$ ,  $d$  a positive integer, and  $s \in \mathbb{R}^d$*

$$\hat{\beta}_r(s) = \int_0^\infty w_r(t) \gamma_t^\circ(s) dt,$$

where  $w_r(t) = \|\gamma_t\|_{\mathcal{L}^2} t^{r/2-1} e^{-t} / \Gamma(r/2)$ .

A straightforward calculation shows that the  $\mathcal{L}^1$ -norm of the weighting function  $w_r$  is the same as the  $\mathcal{L}^1$ -norm of the weighting function  $v_r$  and thus by Theorem [1](#) we get the same upper bound on  $G_0^\circ$ -variation of  $\hat{\beta}_r$ .

**Proposition 5.** *For  $d$  a positive integer and  $r > d/2$ ,*

$$\int_0^\infty w_r(t) dt = k(r, d) \quad \text{and} \quad \|\hat{\beta}_r\|_{G^\circ} \leq \|\hat{\beta}_r\|_{G_0^\circ} \leq k(r, d).$$

Combining Propositions [3](#) and [5](#) with [9](#) we get the next upper bound on rates of approximation of Bessel potentials and their Fourier transforms by Gaussian RBFs.

**Theorem 2.** *For  $d$  a positive integer,  $r > d/2$ , and  $n \geq 1$*

$$\|\beta_r - \text{span}_n G\|_{\mathcal{L}^2} = \|\beta_r - \text{span}_n G_0\|_{\mathcal{L}^2} = \|\hat{\beta}_r - \text{span}_n G_0\|_{\mathcal{L}^2} \leq k(r, d) n^{-1/2}.$$

So for  $r > d/2$  both the Bessel potential of the order  $r$  and its Fourier transform can be approximated by  $\text{span}_n G_0$  with rates bounded from above by  $k(r, d) n^{-1/2}$ . If for some fixed  $c > 0$ ,  $r_d = d/2 + c$ , then with increasing  $d$ ,  $k(r_d, d) = (\pi/2)^{d/4} \Gamma(c/2) / \Gamma(d/2 + c/2)$  converges to zero exponentially fast. So for a sufficiently large  $d$ ,  $\beta_{r_d}$  and  $\hat{\beta}_{r_d}$  can be approximated quite well by Gaussian RBF networks with a small number of hidden units.



## 4 Rates of Approximation by Bessel Potentials

In this section we investigate rates of approximation by linear combinations of translated Bessel potentials.

For  $\tau_x$  the translation operator  $(\tau_x f)(y) = f(y - x)$  and  $r > 0$ , let

$$G_{\beta_r} = \{\tau_x \beta_r \mid x \in \mathbb{R}^d\}$$

denote the *set of translates of the Bessel potential of the order  $r$* . For  $r > d/2$ ,  $G_{\beta_r} \subset \mathcal{L}^2$  since translation does not change the  $\mathcal{L}^2$ -norm.

Let  $r > d/2$ . By a mostly straightforward argument (see [4]), we get

$$\|\hat{\beta}_r\|_{\mathcal{L}^2} = \lambda(r, d) := \pi^{d/4} \left( \frac{\Gamma(r - d/2)}{\Gamma(r)} \right)^{1/2}. \quad (12)$$

The *convolution*  $h * g$  of two functions  $h$  and  $g$  is defined by  $(h * g)(x) = \int_{\mathbb{R}^d} h(y)g(x - y)dy$ . For functions which are convolutions with  $\beta_r$ , the integral representation  $f(x) = \int w(y)\beta_r(x - y)dy$  combined with Theorem 1 gives the following upper bound on  $G_{\beta_r}$ -variation.

**Proposition 6.** *For  $d$  a positive integer and  $r > d/2$ , let  $f = w * \beta_r$ , where  $w \in \mathcal{L}^1$  and  $w$  is continuous a.e.. Then*

$$\|f\|_{G_{\beta_r}} \leq \|f\|_{G_{\beta_r}^\circ} \leq \lambda(r, d)\|w\|_{\mathcal{L}^1}.$$

For  $d$  a positive integer and  $r > d/2$ , consider the *Bessel potential space* (with respect to  $\mathbb{R}^d$ )

$$L^{r,2} = \{f \mid f = w * \beta_r, w \in \mathcal{L}^2\},$$

where  $\|f\|_{L^{r,2}} = \|w\|_{\mathcal{L}^2}$  for  $f = w * \beta_r$ . The function  $w$  has Fourier transform equal to  $(2\pi)^{-d/2} \hat{f}/\hat{\beta}_r$  since the transform of a convolution is  $(2\pi)^{d/2}$  times the product of the transforms, so  $w = (2\pi)^{-d/2}(\hat{f}/\hat{\beta}_r)^\sim$ . In particular,  $w$  is uniquely determined by  $f$  and so the Bessel potential norm is well-defined. The Bessel potential norm is equivalent to the corresponding Sobolev norm (see [1, p. 252]); let  $\kappa = \kappa(d)$  be the constant of equivalence.

For a function  $h : U \rightarrow \mathbb{R}$  on a topological space  $U$ , the *support* of  $h$  is the set  $\text{supp } h = \text{cl}\{u \in U \mid h(u) \neq 0\}$ . It is a well-known consequence of the Cauchy-Schwartz inequality that  $\|w\|_{\mathcal{L}^1} \leq a^{1/2}\|w\|_{\mathcal{L}^2}$ , where  $a = \lambda(\text{supp } w)$ . Hence by Theorem 1 and (12), we have the following estimate.

**Proposition 7.** *Let  $d$  be a positive integer,  $r > d/2$ , and  $f \in L^{r,2}$ . If  $w = (2\pi)^{-d/2}(\hat{f}/\hat{\beta}_r)^\sim$  is bounded, almost everywhere continuous, and compactly supported on  $\mathbb{R}^d$  and  $a = \lambda(\text{supp } w)$ , then  $\|f\|_{G_{\beta_r}^\circ} \leq a^{1/2}\lambda(r, d)\|f\|_{L^{r,2}}$ .*

Combining this estimate of  $G_{\beta_r}$ -variation with Maurey-Jones-Barron's bound (9) gives an upper bound on rates of approximation by linear combinations of  $n$  translates of the Bessel potential  $\beta_r$ .

**Theorem 3.** *Let  $d$  be a positive integer,  $r > d/2$ , and  $f \in L^{r,2}$ . If  $w = (2\pi)^{-d/2}(\hat{f}/\hat{\beta}_r)^\sim$  is bounded, almost everywhere continuous, and compactly supported on  $\mathbb{R}^d$  and  $a = \lambda(\text{supp } w)$ , then for  $n \geq 1$*

$$\|f - \text{span}_n G_{\beta_r}\|_{\mathcal{L}^2} \leq \left(a^{1/2} \lambda(r, d) \|f\|_{L^{2,r}}\right) n^{-1/2}.$$

## 5 Bound on Rates of Approximation by Gaussian RBFs

The results given in previous sections imply an upper bound on rates of approximation by networks with  $n$  Gaussian RBF units for certain functions in the Bessel potential space.

**Theorem 4.** *Let  $d$  be a positive integer,  $r > d/2$ , and  $f \in L^{r,2}$ . If  $w = (2\pi)^{-d/2}(\hat{f}/\hat{\beta}_r)^\sim$  is bounded, almost everywhere continuous, and compactly supported on  $\mathbb{R}^d$  and  $a = \lambda(\text{supp } w)$ , then*

$$\|f - \text{span}_n G\|_{\mathcal{L}^2} \leq \left(a^{1/2} \lambda(r, d) \|f\|_{L^{2,r}} k(r, d)\right) n^{-1/2},$$

$$\text{where } k(r, d) = \frac{(\pi/2)^{d/4} \Gamma(r/2 - d/4)}{\Gamma(r/2)} \text{ and } \lambda(r, d) = \pi^{d/4} \left(\frac{\Gamma(r-d/2)}{\Gamma(r)}\right)^{1/2}.$$

The exponential decrease of  $k(r_d, d)$  mentioned following Theorem 2, with  $r_d = d/2 + c$ , applies also to  $\lambda(r_d, d)$ . Thus unless the constant of equivalence of the Sobolev and Bessel norms  $\kappa(d)$  is growing very fast with  $d$ , even functions with large Sobolev norms will be well approximated for sufficiently large  $d$ . Note that our estimates might be conservative as we used composition of two approximations.

**Acknowledgements.** The collaboration between V. K. and M. S. was partially supported by the 2004-2006 Scientific Agreement among University of Genoa, National Research Council of Italy, and Academy of Sciences of the Czech Republic, V. K. was partially supported by GA ĀR grant 201/05/0557 and by the Institutional Research Plan AV0Z10300504, M. S. by the PRIN Grant from the Italian Ministry for University and Research, project New Techniques for the Identification and Adaptive Control of Industrial Systems; collaboration of P. C. K. with V. K. and M. S. was partially supported by Georgetown University.

## References

1. Adams, R. A., Fournier, J. J. F.: Sobolev Spaces. Academic Press, Amsterdam (2003)
2. Barron, A. R.: Neural net approximation. Proc. 7th Yale Workshop on Adaptive and Learning Systems, K. Narendra, Ed., Yale University Press (1992) 69–72
3. Barron, A. R.: Universal approximation bounds for superpositions of a sigmoidal function. IEEE Transactions on Information Theory **39** (1993) 930–945
4. Carlson, B. C.: Special Functions of Applied Mathematics, Academic Press, New York (1977)

5. Girosi, F.: Approximation error bounds that use VC-bounds. In Proceedings of the International Conference on Neural Networks, Paris (1995) 295-302
6. Girosi, F., Anzellotti, G.: Rates of convergence for radial basis functions and neural networks. In Artificial Neural Networks for Speech and Vision, R. J. Mammone (Ed.), Chapman & Hall, London (1993) 97-113
7. Hartman, E. J., Keeler, J. D., Kowalski, J. M.: Layered neural networks with Gaussian hidden units as universal approximations. *Neural Computation* **2** (1990) 210–215
8. Jones, L. K.: A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *Annals of Statistics* **20** (1992) 608–613
9. Kainen, P. C., Kůrková, V., Sanguineti, M.: Rates of approximation of smooth functions by Gaussian radial-basis- function networks. Research report ICS-976 (2006), [www.cs.cas.cz/research/publications.shtml](http://www.cs.cas.cz/research/publications.shtml).
10. Kon, M. A., Raphael, L. A., Williams, D. A.: Extending Girosi's approximation estimates for functions in Sobolev spaces via statistical learning theory. *J. of Analysis and Applications* **3** (2005) 67-90
11. Kon, M. A., Raphael, L. A.: Approximating functions in reproducing kernel Hilbert spaces via statistical learning theory. Preprint (2005)
12. Kůrková, V.: Dimension-independent rates of approximation by neural networks. In Computer-Intensive Methods in Control and Signal Processing: Curse of Dimensionality, K. Warwick and M. Kárný, Eds., Birkhäuser, Boston (1997) 261–270
13. Kůrková, V.: High-dimensional approximation and optimization by neural networks. Chapter 4 in *Advances in Learning Theory: Methods, Models and Applications*, J. Suykens et al., Eds., IOS Press, Amsterdam (2003) 69–88
14. Kůrková, V., Kainen, P. C., Kreinovich, V.: Estimates of the number of hidden units and variation with respect to half-spaces. *Neural Networks* **10** (1997) 1061–1068
15. Martínez, C., Sanz, M.: *The Theory of Fractional Powers of Operators*. Elsevier, Amsterdam (2001)
16. Mhaskar, H. N.: Versatile Gaussian networks. *Proc. IEEE Workshop of Nonlinear Image Processing* (1995) 70-73
17. Mhaskar, H. N., Micchelli, C. A.: Approximation by superposition of a sigmoidal function and radial basis functions. *Advances in Applied Mathematics* **13** (1992) 350-373
18. Park, J., Sandberg, I. W.: Universal approximation using radial-basis-function networks. *Neural Computation* **3** (1991) 246-257
19. Park, J., Sandberg, I.: Approximation and radial basis function networks. *Neural Computation* **5** (1993) 305-316
20. Pisier, G.: Remarques sur un resultat non publié de B. Maurey. In *Seminaire d'Analyse Fonctionnelle*, vol. I(12). École Polytechnique, Centre de Mathématiques, Palaiseau 1980-1981
21. Stein, E. M.: *Singular Integrals and Differentiability Properties of Functions*. Princeton University Press, Princeton, NJ (1970)
22. Strichartz, R.: *A Guide to Distribution Theory and Fourier Transforms*. World Scientific, NJ (2003)

# Least Mean Square vs. Outer Bounding Ellipsoid Algorithm in Confidence Estimation of the GMDH Neural Networks

Marcin Mrugalski and Józef Korbicz

Institute of Control and Computation Engineering,  
University of Zielona Góra,  
ul. Podgórna 50, 65–246 Zielona Góra, Poland  
{M.Mrugalski, J.Korbicz}@issi.uz.zgora.pl

**Abstract.** The paper deals with the problem of determination of the model uncertainty during the system identification with the application of the Group Method of Data Handling (GMDH) neural network. The main objective is to show how to employ the Least Mean Square (LMS) and the Outer Bounding Ellipsoid (OBE) algorithm to obtain the corresponding model uncertainty.

## 1 Introduction

The scope of applications of mathematical models in the industrial systems is very broad and includes the design of the systems, the control and the system diagnosis [1,2,5,8]. As the most of industrial systems exhibit a non-linear behaviour, this has been the main reason for further development of non-linear system identification theory such as the Artificial Neural Networks (ANNs) [3]. They enable to model the behaviour of complex systems without a priori information about the system structure or parameters. On the other hand, there are no efficient algorithms for selecting structures of classical ANNs and hence many experiments should be carried out to obtain an appropriate configuration. To tackle this problem the GMDH approach can be employed [4]. The synthesis process of GMDH model is based on the iterative processing of a sequence of operations. This process leads to the evolution of the resulting model structure in such a way so as to obtain the best quality approximation of the identified system. The application of the GMDH approach to the model structure selection can improve the quality of the model but it can not eliminate the model uncertainty at all. It follows that the uncertainty of the neural model obtained via system identification can appear during model structure selection and also parameters estimation [9]. In this situation it is necessary to obtain a mathematical description of the neural model uncertainty. The solution to this problem can be application of the Outer Bounding Ellipsoid (OBE) algorithm [6]. In the paper it is shown, how to use the LMS and OBE algorithms to the estimation of the GMDH network parameters in the form of the admissible parameter set called also a parameters uncertainty. This result allows to define the model uncertainty in the form of the confidence interval of the model output.

The paper is organized as follows. Section 2 presents the synthesis of the GMDH network. Section 3 describes the algorithms using during parameters estimation of the

GMDH network. Sections 4 and 5 deal with the problem of the confidence estimation of the neurons via application the LMS and OBE methods, while section 6 presents an example comparing both methods. The final part of this work presents the method of confidence estimation of the whole GMDH network.

## 2 Synthesis of the GMDH Neural Network

The idea of the GMDH approach rely on the replacing of the complex neural model by the set of the hierarchically connected neurons:

$$\tilde{y}_n^{(l)}(k) = \xi \left( (\mathbf{r}_n^{(l)}(k))^T \mathbf{p}_n^{(l)} \right), \quad (1)$$

where  $\tilde{y}_n^{(l)}(k)$  stands for the neuron output ( $l$  is the layer number,  $n$  is the neuron number in the  $l$ -th layer), corresponding to the  $k$ -th measurement of the input  $\mathbf{u}(k) \in \mathbb{R}^{n_u}$  of the system, whilst  $\xi(\cdot)$  denotes a non-linear invertible activation function, i.e. there exists  $\xi^{-1}(\cdot)$ . The model is obtained as a result of the network structure synthesis with the application of the GMDH algorithm [47]:

- 1) Based on the available inputs  $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ , the GMDH network grows its first layer of the neurons. It is assumed that all the possible couples of inputs from signals  $u_1^{(l)}(k), \dots, u_{n_u}^{(l)}(k)$ , belong to the training data set  $\mathcal{T}$ , constitute the stimulation which results in the formation of the neurons outputs  $\tilde{y}_n^{(l)}(k)$ :

$$\tilde{y}_n^{(l)}(k) = f(\mathbf{u}) = f(u_1^{(l)}(k), \dots, u_{n_u}^{(l)}(k)), \quad (2)$$

where  $l$  is the layer number of the GMDH network and  $n$  is the neuron number in the  $l$ -th layer. In order to estimate the unknown parameters of the neurons  $\hat{\mathbf{p}}$  the techniques for the parameter estimation of linear-in-parameter models can be used, e.g. LMS. After the estimation, the parameters are “frozen” during the further network synthesis.

- 2) Using a validation data set  $\mathcal{V}$ , not employed during the parameter estimation phase, calculate a processing error of the each neuron in the current  $l$ -th network layer. The processing error is calculated with the application of the evaluation criterion. Based on the defined evaluation criterion it is possible to select the best-fitted neurons in the layer. The selection methods in the GMDH neural networks plays a role of a mechanism of the structural optimization at the stage of construing a new layer of neurons. During the selection, neurons which have too large value of the evaluation criterion  $Q(\tilde{y}_n^{(l)})$  are rejected. After the selection procedure, the outputs of the selected neurons become the inputs to other neurons in the next layer.
- 3) If the termination condition is fulfilled (the network fits the data with desired accuracy or the introduction of new neurons did not induce a significant increase in the approximation abilities of the neural network), then STOP, otherwise use the outputs of the best-fitted neurons (selected in step 2) to form the input vector for the next layer, and then go to step 1. To obtain the final structure of the network, all unnecessary neurons are removed, leaving only those which are relevant to the computation of the model output. The procedure of removing unnecessary neurons is the last stage of the synthesis of the GMDH neural network.

### 3 Parameters Estimation of the GMDH Neural Network

The main feature of the GMDH algorithm is that the techniques for the parameter estimation of linear-in-parameter models e.g. LMS [4], can be used during the realisation of step  $l$ . It follows from the facts, that the parameters of the each neurons are estimates separately and the neuron's activation function  $\xi(\cdot)$  is invertible, i.e. there exists  $\xi^{-1}(\cdot)$ . The estimation algorithms of linear-in-parameters models requires the system output to be described in the following form:

$$y_n^{(l)}(k) = \left( \mathbf{r}_n^{(l)}(k) \right)^T \mathbf{p}_n^{(l)} + \varepsilon_n^{(l)}(k), \quad (3)$$

and the output error in the case of these algorithms can be defined as:

$$\varepsilon(k)_n^{(l)}(k) = y_n^{(l)}(k) - \hat{y}_n^{(l)}(k). \quad (4)$$

Unfortunately, the application of the LMS to the parameter estimation of neurons (1) is limited by a set of restrictive assumptions. One of them concern the properties of the noise  $\varepsilon$  which affect on the system output  $y(k)$ . In order to obtain the unbiased and minimum variance parameter estimate for (1) it have to be assumed:

$$\mathcal{E} \left[ \varepsilon_n^{(l)} \right] = 0, \quad (5)$$

and

$$\text{cov} \left[ \varepsilon_n^{(l)} \right] = \left( \sigma_n^{(l)} \right)^2 \mathbf{I}. \quad (6)$$

The assumption (5) means that there are no deterministic disturbances, which unfortunately usually are caused by the structural errors. However the condition (6) means that the model uncertainty is described in a purely stochastic way. The assumptions (5) and (6) are not usually fulfill in practice, which cause increasing of the model uncertainty. Opposite to LMS in the more realistic approach is to assume that the errors (4) lie between given prior bounds. This leads directly to the bounded error set estimation class of algorithms, and one of them namely the Outer Bounding Ellipsoid (OBE) algorithm [6] can be employed to obtain the parameter estimate  $\hat{\mathbf{p}}_n^{(l)}(k)$ , as well as an associated parameter uncertainty in the form of the admissible parameter set  $\mathbb{E}$ . In order to simplify the notation the index  $n^{(l)}$  is omitted. Let's assume that  $\varepsilon(k)$  is bounded as:

$$\varepsilon^m(k) \leq \varepsilon(k) \leq \varepsilon^M(k), \quad (7)$$

where  $\varepsilon^m(k)$  and  $\varepsilon^M(k)$  ( $\varepsilon^m(k) \neq \varepsilon^M(k)$ ) are known *a priori*. The expressions (4) and (7) associated with  $k$ -th measurement can be put into the standard form:

$$-1 \leq \underline{y}(k) - \underline{\hat{y}}(k) \leq 1, \quad (8)$$

where:

$$\underline{y}(k)(k) = \frac{2y_z(k) - e_y^M(k) - e_y^m(k)}{e_y^M(k) - e_y^m(k)}, \quad \underline{\hat{y}}(k) = \frac{2}{e_y^M(k) - e_y^m(k)} y_m(k). \quad (9)$$

Let  $\mathbb{S}$  be a strip in parameters space  $\mathbb{E}$ , bounded by two parallel hyperplanes defined as:

$$\mathbb{S}(k) = \{\mathbf{p} \in \mathbb{R}^{n_p} : -1 \leq \underline{y}(k) - \hat{y}(k) \leq 1\} \quad (10)$$

In a recursive OBE algorithm, the data are taken into account one after the other to construct a succession of ellipsoids containing all values of  $\mathbf{p}$  consistent with all previous measurements. After the first  $k$  observations the set of feasible parameters is characterized by the ellipsoid:

$$\mathbb{E}(\hat{\mathbf{p}}(k), \mathbf{P}(k)) = \left\{ \mathbf{p} \in \mathbb{R}^{n_p} : (\mathbf{p} - \hat{\mathbf{p}}(k))^T \mathbf{P}^{-1}(k) (\mathbf{p} - \hat{\mathbf{p}}(k)) \leq 1 \right\}, \quad (11)$$

where  $\hat{\mathbf{p}}(k)$  is the center of the ellipsoid constituting  $k$ -th parameter estimate, and  $\mathbf{P}(k)$  is a positive-definite matrix which specifies its size and orientation. By means of an intersection of the strip (10) and the ellipsoid (11), a region of possible parameter estimates is obtained. This region is outerbounded by a new  $\mathbb{E}(k+1)$  ellipsoid. The OBE algorithm provides rules for computing  $\mathbf{p}(k)$  and  $\mathbf{P}(k)$  in such a way that the volume of  $\mathbb{E}(\hat{\mathbf{p}}(k+1), \mathbf{P}(k+1))$  is minimized. The center of the last  $n_{\mathcal{T}}$ -th ellipsoid constitutes the resulting parameter estimate while the ellipsoid itself represents the feasible parameter set, thus any parameter vector  $\hat{\mathbf{p}}$  contained in  $\mathbb{E}(n_{\mathcal{T}})$  is a valid estimate of  $\mathbf{p}$ .

## 4 Confidence Estimation of the Neuron Via LMS

In order to obtain the GMDH neural model uncertainty it is necessary to obtain the confidence region of the model output for each neuron (11) in the GMDH network. In the case of the LMS, the knowledge regarding the parameter estimates of the neuron:

$$\hat{\mathbf{p}} = \left[ \mathbf{R}^T \mathbf{R} \right]^{-1} \mathbf{R}^T \mathbf{y}, \quad (12)$$

allows obtaining the confidence region of  $\mathbf{p}$  with the confidence level of  $(1 - \alpha)$ :

$$\hat{p}_i - t_{\alpha, n_{\mathcal{D}} - n_p - 1} \sqrt{\hat{\sigma}^2 c_{ii}} < p_i < \hat{p}_i + t_{\alpha, n_{\mathcal{D}} - n_p - 1} \sqrt{\hat{\sigma}^2 c_{ii}}, \quad i = 1, \dots, n_p, \quad (13)$$

where  $c_{ii}$  represents  $i$ -th diagonal element of the matrix  $\mathbf{C} = (\mathbf{R}^T \mathbf{R})^{-1}$ ,  $t_{\alpha, n_{\mathcal{D}} - n_p - 1}$  represents  $(1 - \alpha)$ -th order quantel of a random variable which has a T-Student distribution with  $(n_{\mathcal{D}} - n_p - 1)$  degrees of freedom,  $\hat{\sigma}^2$  represent a variance of the random variable defined as a difference of the system output and its estimate  $y(k) - \hat{y}(k)$ :

$$\hat{\sigma}^2 = \frac{1}{n_{\mathcal{D}} - n_p - 1} \sum_{k=1}^{n_{\mathcal{D}}} (y(k) - \hat{y}(k))^2 = \frac{\mathbf{y}^T \mathbf{y} - \hat{\mathbf{p}}^T \mathbf{R}^T \mathbf{y}}{n_{\mathcal{D}} - n_p - 1}. \quad (14)$$

Finally, the confidence region of the parameters  $\mathbf{p}$  with the confidence level of  $(1 - \alpha)$  can be defined as follows:

$$(\hat{\mathbf{p}} - \mathbf{p})^T \mathbf{R}^T \mathbf{R} (\hat{\mathbf{p}} - \mathbf{p}) \leq (n_p + 1) \hat{\sigma}^2 F_{\alpha, n_{\mathcal{D}} - n_p - 1}^{n_p + 1}, \quad (15)$$

where  $F_{\alpha, n_{\mathcal{D}} - n_p - 1}^{n_p + 1}$  is  $(1 - \alpha)$ -th order quantel of a random variable which has a Snedecor's  $F$ -Distribution with  $(n_{\mathcal{D}} - n_p - 1)$  and  $(n_p + 1)$  degrees of freedom. In order to obtain the  $(1 - \alpha)$  confidence interval for the neuron output (11) it is necessary

to assume that  $\hat{y}(k)$  is a random variable, which has a Gaussian distribution. Then expected value has the following form:

$$\mathcal{E}[y(k)] = \mathcal{E}[\mathbf{r}^T(k)\hat{\mathbf{p}}] = \mathbf{r}^T(k)\mathcal{E}[\hat{\mathbf{p}}] = \mathbf{r}^T(k)\mathbf{p}, \quad (16)$$

and the variance is:

$$\text{var}[y(k)] = \mathbf{r}^T(k)\mathcal{E}[(\hat{\mathbf{p}} - \mathbf{p})(\hat{\mathbf{p}} - \mathbf{p})^T]\mathbf{r}(k). \quad (17)$$

Taking into consideration, that:  $\mathcal{E}[(\hat{\mathbf{p}} - \mathbf{p})(\hat{\mathbf{p}} - \mathbf{p})^T] = (\mathbf{R}^T\mathbf{R})^{-1}\sigma^2$ , the (17) has the following form:

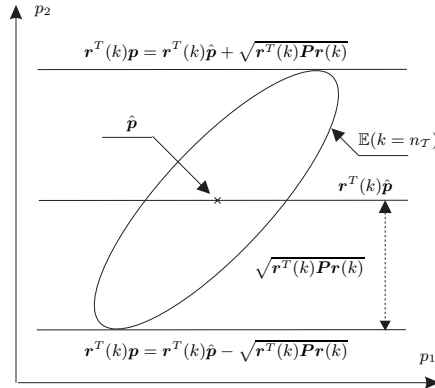
$$\text{var}[y(k)] = \mathbf{r}^T(k)(\mathbf{R}^T\mathbf{R})^{-1}\mathbf{r}(k)\sigma^2. \quad (18)$$

Finally, the  $(1 - \alpha)$  confidence interval for the neuron output (11) has the following form:

$$\begin{aligned} \hat{y}(k) - t_{\alpha, n_{\mathcal{D}} - n_p - 1} \sqrt{\hat{\sigma}^2 \mathbf{r}^T(k) (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{r}(k)} &< \mathbf{r}^T(k) \mathbf{p} < \\ \hat{y}(k) + t_{\alpha, n_{\mathcal{D}} - n_p - 1} \sqrt{\hat{\sigma}^2 \mathbf{r}^T(k) (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{r}(k)}. \end{aligned} \quad (19)$$

## 5 Confidence Estimation of the Neuron Via OBE

In the case of the OBE algorithm the range of the confidence interval of the neuron output depends on the size and the orientation of the ellipsoid which define the admissible parameter set  $\mathbb{E}$  (cf. Fig 1). Taking the minimal and maximal values of the admissible



**Fig. 1.** Relation between the size of the ellipsoid and the neuron output uncertainty

parameter set  $\mathbb{E}$  into consideration it is possible to determine the minimal and maximal values of confidence interval of the neuron output:

$$\mathbf{r}^T(k)\hat{\mathbf{p}} - \sqrt{\mathbf{r}^T(k)\mathbf{P}\mathbf{r}(k)} \leq \mathbf{r}^T(k)\mathbf{p} \leq \mathbf{r}^T(k)\hat{\mathbf{p}} + \sqrt{\mathbf{r}^T(k)\mathbf{P}\mathbf{r}(k)}. \quad (20)$$



## 6 An Illustrative Example

The purpose of the present section is to show the effectiveness of the proposed approaches based on the LMS and OBE in the task of parameters estimation of the neurons in the GMDH network. Let us consider the following static system:

$$y(k) = p_1 \sin(u_1^2(k)) + p_2 u_2^2(k) + \varepsilon(k), \quad (21)$$

where the nominal values of parameters are  $\mathbf{p} = [0.5, -0.2]^T$ , the input data  $\mathbf{u}(k)$  and the noise  $\varepsilon(k)$ ,  $k = 1, \dots, n_T$  are generated according to the uniform distribution, i.e.  $\mathbf{u}(k) \in \mathcal{U}(0, 2)$  and  $\varepsilon(k) \in \mathcal{U}(-0.05, 0.1)$ . Note that the noise does not satisfy (5). The problem is to obtain the parameter estimate  $\hat{\mathbf{p}}$  and the corresponding neuron uncertainty using the set of input-output measurements  $\{(u(k), y(k))\}_{k=1}^{n_T=20}$ . To tackle this task, the approaches described in Sections 3-5 were employed. In the case of the LMS the parameters uncertainty was obtained with the confidence level of 95%, whereas the application of the OBE allowed to calculate the parameters uncertainty with the confidence level of 100%. The minimal and maximal values of the parameter estimates for the both methods are presented in the table 1. The results show that the parameters es-

**Table 1.** Parameters and their uncertainty obtained with the application of the LMS and OBE

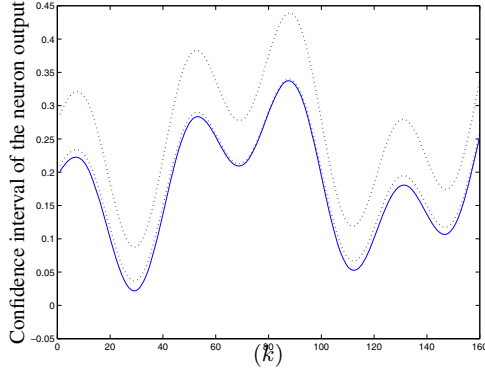
$\mathbf{p}$	OBE $\hat{\mathbf{p}}$	OBE $[p^{\min}, p^{\max}]$	LMS $\hat{\mathbf{p}}$	LMS $[p^{\min}, p^{\max}]$
$p_1$	0.4918	[0.4692, 0.5144]	0.5224	[0.4617, 0.5831]
$p_2$	-0.1985	[-0.2039, -0.1931]	-0.1768	[-0.1969, -0.1567]

timates obtained with the application of the OBE are similar to the nominal parameters  $\mathbf{p} = [0.5, -0.2]^T$ , opposite to parameters estimates calculated with the LMS. It follows from the fact that the condition (5) concerning noise is not fulfilled. The achieved regions of possible parameter estimate allow to obtain the neuron uncertainty in the form of the confidence interval of the model output. For both methods the intervals are calculated for the following common validation signal:

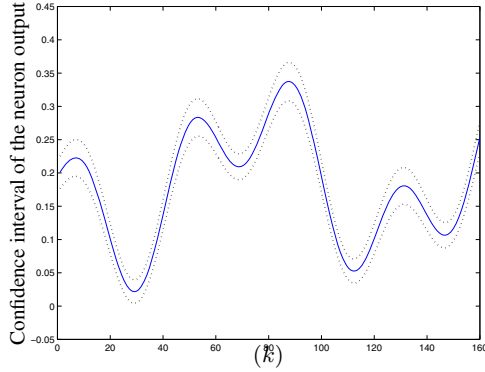
$$u_1(k) = -0.1 \sin(0.02\pi(k)) + 0.2 \sin(0.05\pi(k)) + 0.8 \quad \text{for } k = 1, \dots, 160,$$

$$u_2(k) = 0.25 \sin(2\pi(k/100)) + 0.1 \sin(\pi(k/20)) + 1.0 \quad \text{for } k = 1, \dots, 160.$$

In the case of the LMS the confidence interval of the model output (Fig. 2) was calculated with the application of expression (19). Figure 3 shows the confidence interval of the model output obtained with the application of the OBE based on the expression (20). The results obtained with the LMS indicate that the neuron output uncertainty interval does not contain the system output calculated based on the nominal parameters  $\mathbf{p}$ . Therefore, only the application of the OBE allows to obtain unbiased parameters estimates and neuron uncertainty.



**Fig. 2.** The confidence interval of the neuron output ( $\cdots$ ) and the system output ( $—$ )



**Fig. 3.** The confidence interval of the neuron output ( $\cdots$ ) and the system output ( $—$ )

## 7 Confidence Estimation of the Whole GMDH Network Via OBE

By using the already proposed the OBE approach, it is possible to obtain only the bounds of the confidence interval of the neuron output in the first layer of the GMDH network. It follows from the fact that the outputs of the selected neurons become the inputs to other neurons in the next layer. So, the output bounds became the bounds of the regressor error in the next layer. From this reason, an error in the regressor should be taken into account during the design procedure of the neurons from the subsequent layers. Let us denote an unknown “true” value of the regressor  $\mathbf{r}_n(k)$  by a difference between a measured value of the regressor  $\mathbf{r}(k)$  and the error in the regressor  $\mathbf{e}(k)$ :

$$\mathbf{r}_n(k) = \mathbf{r}(k) - \mathbf{e}(k), \quad (22)$$

where the regressor error  $\mathbf{e}(k)$  is bounded as follows:

$$-\epsilon_i \leq e_i(k) \leq \epsilon_i, \quad i = 1, \dots, n_p. \quad (23)$$

Substituting (22) into (20) it can be show that the partial models output uncertainty interval have following form:

$$y^m(k)(\hat{\mathbf{p}}) \leq \mathbf{r}^T(k)\mathbf{p} \leq y^M(k)(\hat{\mathbf{p}}), \quad (24)$$

where:

$$y^m(k)(\hat{\mathbf{p}}) = \mathbf{r}_n^T(k)\hat{\mathbf{p}} + \mathbf{e}^T(k)\hat{\mathbf{p}} - \sqrt{(\mathbf{r}_n(k) + \mathbf{e}(k))^T \mathbf{P} (\mathbf{r}_n(k) + \mathbf{e}(k))}, \quad (25)$$

$$y^M(k)(\hat{\mathbf{p}}) = \mathbf{r}_n^T(k)\hat{\mathbf{p}} + \mathbf{e}^T(k)\hat{\mathbf{p}} + \sqrt{(\mathbf{r}_n(k) + \mathbf{e}(k))^T \mathbf{P} (\mathbf{r}_n(k) + \mathbf{e}(k))}. \quad (26)$$

In order to obtain the final form of the expression (24) it is necessary to take into consideration the bounds of the regresor error (23) in the expressions (25) and (26):

$$y^m(k)(\hat{\mathbf{p}}) = \mathbf{r}_n^T(k)\hat{\mathbf{p}} + \sum_{i=1}^{n_p} \text{sgn}(\hat{p}_i)\hat{p}_i\epsilon_i - \sqrt{\bar{\mathbf{r}}_n^T(k)\mathbf{P}\bar{\mathbf{r}}_n(k)}, \quad (27)$$

$$y^M(k)(\hat{\mathbf{p}}) = \mathbf{r}_n^T(k)\hat{\mathbf{p}} + \sum_{i=1}^{n_p} \text{sgn}(\hat{p}_i)\hat{p}_i\epsilon_i + \sqrt{\bar{\mathbf{r}}_n^T(k)\mathbf{P}\bar{\mathbf{r}}_n(k)}, \quad (28)$$

where  $\bar{r}_{n,i}(k) = r_{n,i}(k) + \text{sgn}(r_{n,i}(k))\epsilon_i$ .

## 8 Conclusions

The objective of this paper was to obtain GMDH models and calculate their uncertainty with the application of the LMS and OBE. It was shown how to estimate parameters and the corresponding uncertainty of the particular neuron and the whole GMDH network. The comparison of both methods was done on the illustrative example.

## References

1. Delaleau, E., Louis, J.P., Ortega, R.: Modeling and Control of Induction Motors. *Int. Journal of Applied Mathematics and Computer Science*. **11** (2001) 105–129
2. Etien, E., Cauet, S., Rambault, L., Champenois, G.: Control of an Induction Motor Using Sliding Mode Linearization. *Int. Journal of Applied Mathematics and Computer Science*. **12** (2001) 523–531
3. Gupta, M.M., Liang, J., Homma, N.: *Static and Dynamic Neural Networks*, John Wiley & Sons, New Jersey (2003)
4. Ivakhnenko, A.G., Mueller, J.A.: *Self-organizing of Nets of Active Neurons. System Analysis Modelling Simulation*. **20** (1996) 93–106
5. Korbicz, J., Kościelny, J.M., Kowalczyk, Z., Cholewa, W. (eds.): *Fault Diagnosis. Models, Artificial Intelligence, Applications*, Springer-Verlag, Berlin (2004)
6. Milanese, M., Norton, J., Piet-Lahanier, H., Walter, E. (eds.): *Bounding Approaches to System Identification*, Plenum Press, New York (1996)
7. Mrugalski, M.: *Neural Network Based Modelling of Non-linear Systems in Fault Detection Schemes*, Ph.D. Thesis, University of Zielona Góra, Zielona Góra (2004) (In Polish)
8. Witczak, M.: *Advances in Model-based Fault Diagnosis with Evolutionary Algorithms and Neural Networks*. *Int. Journal of Applied Mathematics and Computer Science*. **16** (2006) 85–99
9. Witczak, M., Korbicz, J., Mrugalski, M., Patton, R.J.: A GMDH neural network based approach to robust fault detection and its application to solve the DAMADICS benchmark problem. *Control Engineering Practice*. **14** (2006) 671–683

# On Feature Extraction Capabilities of Fast Orthogonal Neural Networks

Bartłomiej Stasiak and Mykhaylo Yatsymirskyy

Institute of Computer Science, Technical University of Łódź  
ul. Wólczajska 215, 93-005 Łódź, Poland  
{basta, jacym}@ics.p.lodz.pl

**Abstract.** The paper investigates capabilities of fast orthogonal neural networks in a feature extraction task for classification problems. Neural networks with an architecture based on the fast cosine transform, type II and IV are built and applied for extraction of features used as a classification base for a multilayer perceptron. The results of the tests show that adaptation of the neural network allows to obtain a better transform in the feature extraction sense as compared to the fast cosine transform. The neural implementation of both the feature extractor and the classifier enables integration and joint learning of both blocks.

## 1 Introduction

One of the most crucial stages in pattern recognition and classification tasks is feature extraction. The risk and challenge involved here is the necessity of substantial data reduction, minimizing noise and within-class pattern variability, while retaining the data essential from the classification point of view. Having insight into the nature of the analysed data we can develop some well suited methods, effective but applicable to a strictly confined group of problems, like e.g. human fingerprint matching. A different approach involves using general tools known for their optimal properties in the data-packing sense, such as principal component analysis (PCA), discrete cosine transform (DCT), wavelet or Fourier descriptors [1,2,3]. These tools are also often adapted for a specific type of tasks which usually leads to lowering their generality. For example, shape-adaptive DCT (SA-DCT) [4] performs well in image coding but its results depend significantly on the preceding segmentation stage.

The general disadvantage of problem-specific methods is that they often need substantial human intelligence contribution to the feature extraction process. The statistical methods, such as PCA, usually offer more automatic operation at the cost of greater computational complexity. The existence of fast orthogonal transforms algorithms, such as fast cosine transform (FCT) or fast Fourier transform (FFT) allows, however, to lower the complexity necessary to obtain a relatively small amount of information-rich features.

The question is if the most important data from the compression point of view is actually best suited for classification purposes. It is a common approach to use

low-frequency spectral coefficients minimizing the mean square reconstruction error [25]. However, the class-relevant information may be also scattered over a wider spectrum range and more features are then needed for proper classification. Replacing the Fourier or cosine feature extractor with a different transform, better suited for the specific problem, would help reduce the input passed to the classifier module or enhance the recognition results for the same input size. Ideally, such a transform should easily adapt itself to the given training dataset and it should offer a fast computational procedure in both adaptation and forward processing stages.

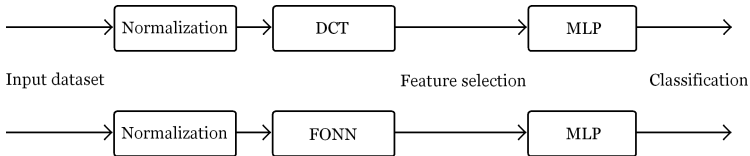
In the previous papers a concept of linear neural networks with sparse architecture based on fast orthogonal transforms algorithms has been introduced [6]. The networks of this type combine the weights adaptation and learning capabilities with fast computational scheme and data compression property specific to the underlying orthogonal transforms. The orthogonality itself was used in [7] to substantially reduce the number of adapted weights by means of basic operation orthogonal neuron (BOON) introduction.

In this paper we consider application of a fast orthogonal neural network proposed in [7] as a feature extractor for a nonlinear multilayer perceptron performing the data classification task. We take advantage of the structural homogeneity of both networks allowing seamless data processing, gradient computation and error backpropagation. The classification results are compared to those obtained with a standard, non-adaptable fast cosine transform block.

## 2 Classification Framework

Let us consider a dataset  $A = \{x_s\}_{s=0, \dots, S-1}$ , where  $x_s \in \mathfrak{R}^N$ . Each of the vectors  $x_s$  belongs to one of the  $L$  classes and the membership function  $f : A \rightarrow C = \{c_1, \dots, c_L\}$  is known. The dataset is divided into two disjoint subsets  $A = A_{train} \cup A_{test}$ . Our goal is to discover  $f|_{A_{test}}$  assuming the knowledge of  $f|_{A_{train}}$ .

The general classification system comprises three basic blocks performing normalization, feature extraction and classification. As we consider two different approaches to the construction of the second block, two variants of the general system have been set up (Fig. 1). The first one performs a standard discrete cosine transform of the normalized data (DCT), being a comparison base for the second one which uses an adaptable fast orthogonal neural network (FONN).



**Fig. 1.** Two variants of processing. DCT feature extractor (top) and adaptable feature extractor (bottom).

Both variants were implemented twice and two independent experiments were made: the first one with discrete cosine transform, type II (DCT2) and the second one with discrete cosine transform, type IV (DCT4).

## 2.1 Normalization

As we do not impose any additional constraints on the nature of the analysed data, the only operations involved here consist of the constant component subtraction and vector length normalization performed for every  $x_s \in A$ .

## 2.2 Feature Extraction

Each normalized input vector  $x$  is subjected to discrete cosine transforms given as [8]:

$$L_N^{II}(k) = \text{DCT}_N^{II} \{x(n)\} = \sum_{n=0}^{N-1} x(n) C_{4N}^{(2n+1)k}, \quad (1)$$

$$L_N^{IV}(k) = \text{DCT}_N^{IV} \{x(n)\} = \sum_{n=0}^{N-1} x(n) C_{8N}^{(2n+1)(2k+1)}, \quad (2)$$

where  $n, k = 0, 1, \dots, N-1$ ;  $C_K^r = \cos(2\pi r/K)$ .

The actual computations are performed using homogeneous two-stage algorithms of fast cosine transform, type II and IV with tangent multipliers (mFCT2, mFCT4) [9,10] depicted in the form of directed graphs in Fig. 2, 3.

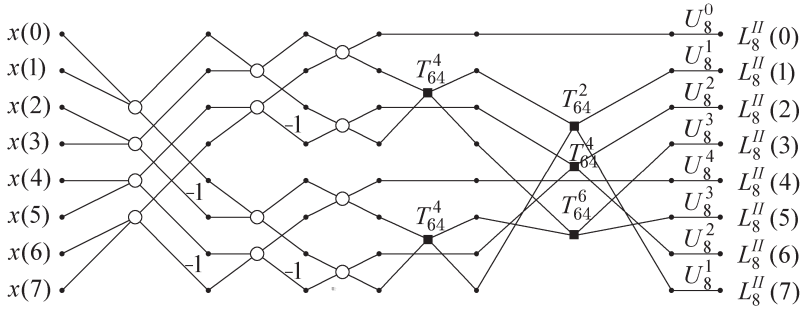
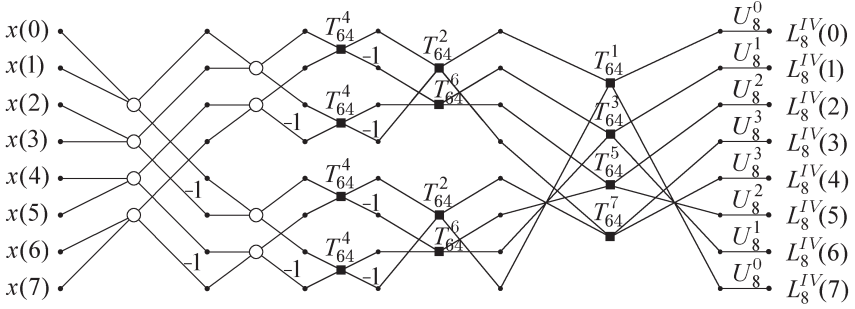


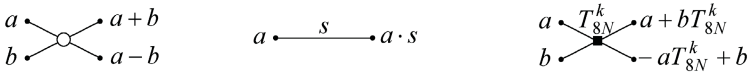
Fig. 2. Directed graph of the mFCT2 algorithm for  $N = 8$

The basic operations are shown in Fig. 4 and the values of  $U_N^k$  are defined recursively for mFCT2:

$$\begin{aligned} U_N^0 &= 1, U_N^{N/2} = \sqrt{2}/2, \\ U_4^1 &= \sqrt{2}/2 \cdot C_{16}^1, U_K^k = U_{K/2}^k C_{4K}^k, U_K^{K/2-k} = U_{K/2}^k C_{4K}^{K/2-k}, \\ k &= 1, 2, \dots, K/4 - 1, U_K^{K/4} = \sqrt{2}/2 \cdot C_{4K}^{K/4}, K = 8, 16, \dots, N, \end{aligned} \quad (3)$$



**Fig. 3.** Directed graph of the mFCT4 algorithm for  $N = 8$



**Fig. 4.** Basic operations of the mFCT2 algorithm

and for mFCT4 accordingly:

$$\begin{aligned}
 U_K^k &= U_{K/2}^k C_{8K}^{2k+1}, \\
 U_2^0 &= C_{16}^1, U_K^{K/2-1-k} = U_{K/2}^k C_{8K}^{K-2k-1}, \\
 k &= 0, 1, \dots, K/4 - 1, K = 4, 8, \dots, N.
 \end{aligned}
 \tag{4}$$

The mFCT2 algorithm (Fig. 2) is used in two ways. Firstly, it is directly applied to compute the DCT2 coefficients of the normalized input vectors. Some of the coefficients are then selected and passed on to the classifier block. This constitutes the first processing variant (Fig. 1). In the second variant the mFCT2 block is implemented as a fast orthogonal neural network. Exactly the same procedure for both variants is then applied with the mFCT4 algorithm (Fig. 3).

### 2.3 Fast Orthogonal Neural Network Construction

The diagrams in Fig. 2, 3 serve as a starting point for the fast orthogonal neural networks design. They contain nodes with two inputs and two outputs, grouped into several layers, representing basic arithmetic operations on the processed data. Each basic operation may be presented in the form of multiplication:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = P \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix},
 \tag{5}$$

where the elements of the matrix  $P$  depend on the type of the operation.

Taking the orthogonality of the nodes into account leads to the definition of two basic forms of the matrix  $P$ :

$$P_2 = \begin{bmatrix} u & w \\ -w & u \end{bmatrix}, P_1 = \begin{bmatrix} 1 & t \\ -t & 1 \end{bmatrix}.
 \tag{6}$$

The matrices  $P_2$ ,  $P_1$  offer a two-fold and a four-fold free coefficients reduction, accordingly, as compared to an unconstrained, possibly non-orthogonal 2x2 matrix. In both the presented mFCT algorithms with tangent multipliers the most efficient  $P_1$  matrix may be applied in the second stage of the transform. The first stage may be constructed with either of the matrices  $P_1$ ,  $P_2$  or it may be implemented directly.

Constructing the neural network architecture, the operations  $P$  are converted into neurons. On account of their specificity, expressed in the presence of two interconnected outputs, a notion of a basic operation orthogonal neuron (BOON) was proposed in [7] along with formulas for weights adaptation and error back-propagation:

$$\begin{bmatrix} \frac{\partial E}{\partial u} \\ \frac{\partial E}{\partial w} \end{bmatrix} = \begin{bmatrix} v_1 & v_2 \\ v_2 & -v_1 \end{bmatrix} \cdot \begin{bmatrix} e_1^{(n)} \\ e_2^{(n)} \end{bmatrix}, \quad (7)$$

$$\begin{bmatrix} e_1^{(n-1)} \\ e_2^{(n-1)} \end{bmatrix} = P_2^T \cdot \begin{bmatrix} e_1^{(n)} \\ e_2^{(n)} \end{bmatrix}, \quad (8)$$

$$\frac{\partial E}{\partial t} = [v_2, -v_1] \cdot \begin{bmatrix} e_1^{(n)} \\ e_2^{(n)} \end{bmatrix}, \quad (9)$$

$$\begin{bmatrix} e_1^{(n-1)} \\ e_2^{(n-1)} \end{bmatrix} = P_1^T \cdot \begin{bmatrix} e_1^{(n)} \\ e_2^{(n)} \end{bmatrix}, \quad (10)$$

where  $v_1$  and  $v_2$  are the inputs of the basic operation, the vector  $[e_1^{(n)}, e_2^{(n)}]^T$  refers to error values propagated back from the next layer and the vector  $[e_1^{(n-1)}, e_2^{(n-1)}]^T$  defines the error values to propagate back to the previous one. We use the formulas (7) - (10) to compute gradient values for every basic operation of the orthogonal network. In this way we apply a sparse approach to the learning process as opposed to operations on full matrices known from classical multilayer networks [111].

## 2.4 Feature Selection and Classification

One of the consequences of the architectural design of a fast orthogonal network is that one can preset its weights so that it initially performs accurately the underlying transform. As we assume that the most important information is contained in the first  $K$  DCT coefficients (not including the constant component in the case of the DCT2), we can similarly consider the first  $K$  outputs of the orthogonal network as the feature selection strategy. Their values are actually equal at the beginning for both processing variants. The difference lies in their adaptation during the learning process in the second variant.

The classification block consists of a multilayer perceptron with one hidden non-linear layer with unipolar sigmoid activation function. The size of the hidden layer is chosen experimentally. The number of inputs and outputs is dependent on the number of the selected DCT coefficients and the number of classes in the



dataset, respectively. Each output corresponds to one of the classes  $c_1, \dots, c_L$  and the winner-takes-all strategy is used for the resulting class determination [10, 11].

The classifier is the only adaptable part in the first variant of the system. The conjugate gradient method is used for weights adaptation and the gradient is determined by means of error backpropagation [10, 11]. The error is defined as:

$$E = \frac{1}{S \cdot L} \sum_{s=0}^{S-1} \sum_{l=1}^L (y_s(l) - d_s(l))^2, \quad (11)$$

where

$$d_s(l) = \begin{cases} 1, & \text{if } f(x_s) = c_l \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

In the second variant of the system the error signal is propagated back also from the hidden layer, just as if there were one hidden layer more before the existing one. This signal is sent back to the orthogonal network so that it can adapt its weights. In this way both networks actually constitute one hybrid neural network with two different architectures. Moreover, the conjugate gradient algorithm is basically unaware of any differences as it operates on vectors containing the gradient and weights values of both parts.

### 3 Testing Material and Simulation Results

The Synthetic Control Chart Time Series dataset [12] has been selected as the test base. It contains 600 time series of 60 real numbers each, divided into six classes varied by the character of the random process used for generating the given sequence (normal, cyclic, increasing trend, decreasing trend, upward shift, downward shift).

We split the whole datasets in two subsets  $A_{train}$  containing 420 sequences and  $A_{test}$  containing 180 sequences. Each sequence was extended from 60 to 64 elements by adding four zero elements to the end.

#### 3.1 Testing Procedure

All the tests were performed with the same neural network parameters, chosen experimentally after some preliminary tests. The number of epochs was set to 200 and 8 hidden neurons were used in the MLP block. The only varying parameter was the number of the classifier inputs.

Since the main purpose of the tests was to determine the fitness of the fast orthogonal network for feature extraction enhancement, we employed a special teaching procedure consisting of two phases. In the first phase (epochs 1-100) only the MLP part was adapted in both processing variants. The weights of the orthogonal network were fixed to such values that it realized the DCT transform. In the second phase (epochs 101-200) the classifier alone continued to adapt its weights in the first processing variant, while in the second one the adaptation of the orthogonal network was also turned on. The weights of the MLP part were initialized with random values from the range  $[-1, 1]$  in both variants.

### 3.2 Tests Results and Analysis

The classification results for both processing variants are presented in Tables [1](#), [2](#) respectively. The teaching was repeated 30 times for every table row, i.e. for every number of features  $K$ , and the averaged values are displayed.

**Table 1.** Recognition results for DCT feature extractor

K	Error		Recognition rate ( $A_{train}$ )		Recognition rate ( $A_{test}$ )	
	DCT2	DCT4	DCT2	DCT4	DCT2	DCT4
1	0.265	0.317	0.6442856	0.5157144	0.6551851	0.4957407
2	0.182	0.195	0.8576984	0.8430161	0.8164814	0.8533333
3	0.114	0.121	0.9453969	0.9484127	0.9318518	0.9370371
4	0.050	0.081	0.9903119	0.9731526	0.9777777	0.9557472

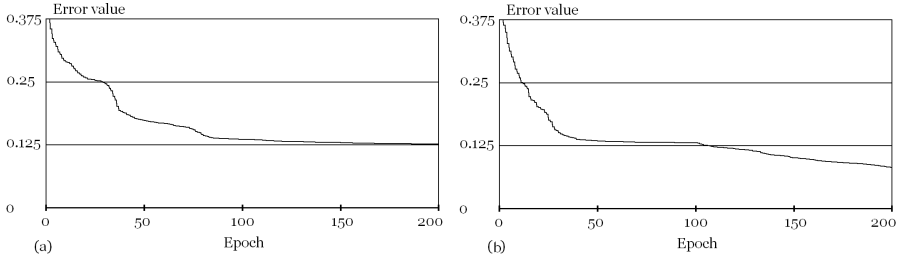
**Table 2.** Recognition results for orthogonal neural network

K	Error		Recognition rate ( $A_{train}$ )		Recognition rate ( $A_{test}$ )	
	DCT2	DCT4	DCT2	DCT4	DCT2	DCT4
1	0.217	0.268	0.8042064	0.7347619	0.7574074	0.6564815
2	0.126	0.139	0.9396032	0.9383334	0.8796297	0.9057408
3	0.037	0.072	0.9970634	0.9765079	0.9616667	0.9429631
4	0.010	0.051	0.9993651	0.9804762	0.9840740	0.9444444

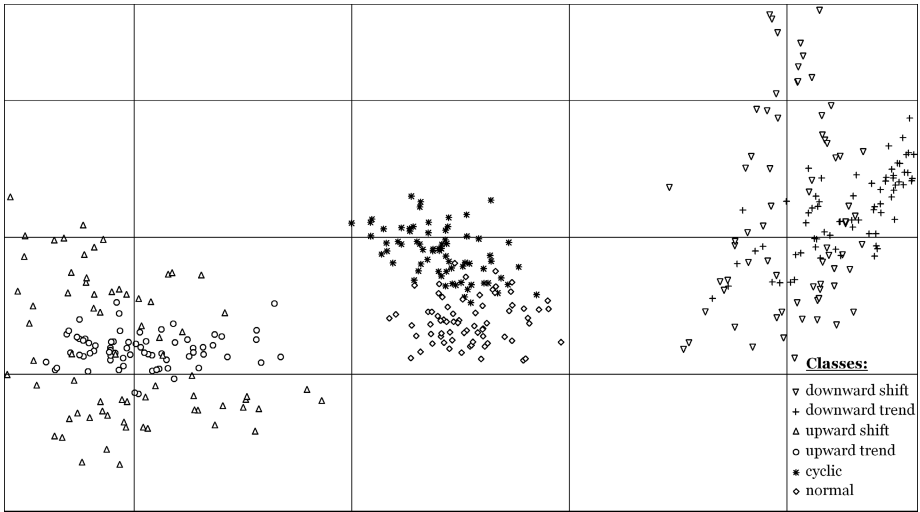
The first conclusion is that for the presented dataset the DCT2 transform performs better than DCT4, both in the fixed variant (Tab. [1](#)) and in the adapted one (Tab. [2](#)), which is reflected in the lower error values and higher recognition rates. For both transforms we can observe that four DCT coefficients contain enough information to enable successful recognition of almost all the investigated samples (recognition rate over 95%). The most interesting observation, however, is the substantial increase in the recognition rate for the orthogonal network in case of insufficient number of features ( $K = 1, 2, 3$ ).

It is clear that adapting the weights of the orthogonal network enhances the classification potential of its output values with respect to the DCT coefficients. This may also be observed during the second phase of the learning process when, after teaching the classifier, the FONN block adaptation is turned on. The error, which usually reaches a plateau at the end of the first phase, starts decreasing again in the second one. The typical error curves are shown in Fig. [5](#).

Figures [6](#), [7](#) further illustrate the capabilities of the fast orthogonal network. All the data vectors from  $A_{train}$  are presented in the feature space of the two lowest DCT2 components. It may be seen (Fig. [6](#)) that however the DCT2 clearly distinguishes three subsets comprising two classes each, it fails to perform proper class separation within the subsets. Adapting the transform (Fig. [7](#)) leads to



**Fig. 5.** Error curves for DCT+MLP training (a) and for FNN+MLP training (b)

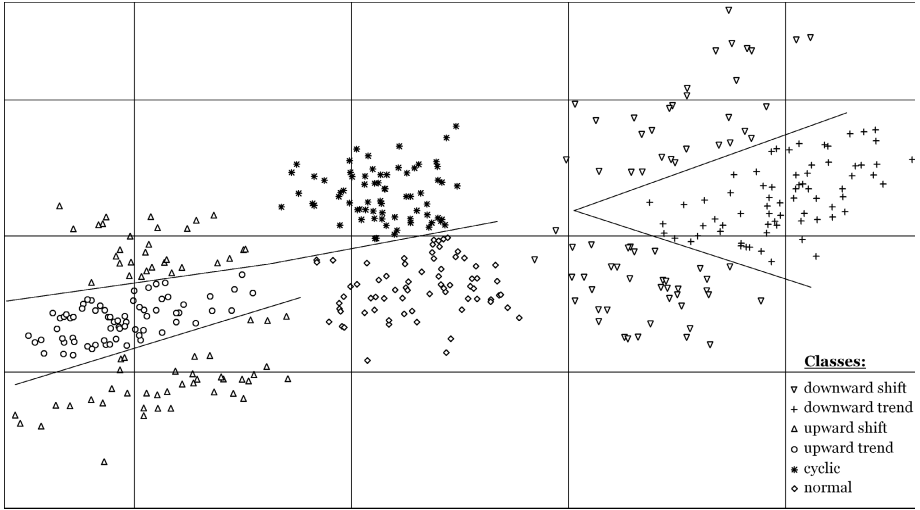


**Fig. 6.** Data points in the 2-dimensional feature space for DCT2+MLP

obtaining more separated distributions of data points (two of them bimodal) at the expense of reducing the distance between the three subsets.

Considering practical applications there are two questions that must be taken into account. Firstly, the orthogonal network may be more prone to increase the generalization error as it does not confine its outputs strictly to the low-frequency spectral components. It is particularly clear in the DCT4  $A_{test}$  result for  $K = 4$  which is even worse in the adapted case, although for the  $A_{train}$  it is still better than in the fixed one. A validation mechanism should be therefore included in the learning process.

The second issue concerns the necessity of processing the data by the FNN block during the learning process. In the first variant the DCT coefficients may be computed once and then used for classifier training. Using the neural feature extractor means that the data must be processed during every epoch to allow its weights adaptation.



**Fig. 7.** Data points in the 2-dimensional feature space for FONN+MLP. The separating hyperplanes have been marked manually.

There are many factors potentially influencing the overall efficiency, such as the type and implementation of the orthogonal transform or the dataset characteristics. Obviously, we would not achieve much on small and medium sized datasets as the presented one. However, in high dimensional computer vision and image recognition problems where the number and location of the crucial spectral components is difficult to determine, the possibility of fine-tuning the feature extractor or even reducing the feature space would be desirable.

## 4 Conclusion

The presented results show the superiority of the adaptable fast orthogonal neural networks over the fast cosine transform, type II and IV in the feature extraction task. The analysed networks proved to be able to concentrate more classification-relevant information in the same number of coefficients. It seems justified to state that neural networks of this type may help reduce the dimensionality of the feature space in complex classification problems.

The properties of the presented networks, i.a. the classification capabilities, the recursive character of the neural connections architecture and easy integration with other neural systems based on gradient adaptation techniques are opening interesting possibilities of further research. Future works will explore the applications of fast orthogonal neural networks to real problems in the field of signal classification.

## References

1. Osowski, S.: Neural networks for information processing. (in Polish) OWPW, Warsaw (2000)
2. Osowski, S., Nghia, D.D.: Fourier and wavelet descriptors for shape recognition using neural networks - a comparative study. *Pattern Recognition* **35** (2002) 1949–1957
3. Stasiak, B., Yatsymirskyy, M.: Application of Fourier-Mellin Transform To Categorization of 3D Objects. (in Polish) In: Proc. of the III Conference on Information Technologies, Gdańsk, Poland (2005)
4. Sikora, T., Makai, B.: Shape-adaptive DCT for generic coding of video. *IEEE Trans. on Circuits Syst. Video Technol.* **5** (1995) 59–62
5. Pan, Z., Rust, A., Bolouri, H.: Image Redundancy Reduction for Neural Network Classification using Discrete Cosine Transforms. In: Proc. of the International Joint Conference on Neural Networks, Como, Italy **3** (2000) 149–154
6. Jacymirski, M., Szczepaniak, P.S.: Neural realization of fast linear filters. In: Proc. of the 4th EURASIP - IEEE Region 8 International Symposium on Video/Image Processing and Multimedia Communications. (2002) 153-157
7. Stasiak, B., Yatsymirskyy, M.: Fast orthogonal neural networks. In: Proc. of the 8th International Conf. on Artificial Intelligence and Soft Computing. (2006) 142-149
8. Rao, K.R., Yip, P.: Discrete cosine transform. Academic Press, San Diego (1990)
9. Jacymirski, M.: Fast homogeneous algorithms of cosine transforms, type II and III with tangent multipliers. (in Polish) *Automatics* **7** AGH University of Science and Technology Press, Cracow (2003) 727-741
10. Stasiak, B., Yatsymirskyy, M.: Recursive learning of fast orthogonal neural networks. In: Proc. of the International Conf. on Signals and Electronic Systems. (2006) 653-656
11. Rutkowski, L.: Methods and techniques of artificial intelligence. (in Polish) Polish Scientific Publishers PWN (2005)
12. Hettich, S., Bay, S.D.: The UCI KDD Archive [<http://kdd.ics.uci.edu>]. Irvine, CA: University of California, Department of Information and Computer Science (1999)

# Neural Computations by Asymmetric Networks with Nonlinearities

Naohiro Ishii<sup>1</sup>, Toshinori Deguchi<sup>2</sup>, and Masashi Kawaguchi<sup>3</sup>

<sup>1</sup> Aichi Institute of Technology, Yakusacho, Toyota 470-0392, Japan  
ishii@aitech.ac.jp

<sup>2</sup> Gifu National College of Technology, Motosu, Gifu 501-0495, Japan  
deguchi@gifu-nct.ac.jp

<sup>3</sup> Suzuka National College of Technology, Suzuka, Mie 510-0294, Japan  
masashi@elec.suzuka-ct.ac.jp

**Abstract.** Nonlinearity is an important factor in the biological visual neural networks. Among prominent features of the visual networks, movement detections are carried out in the visual cortex. The visual cortex for the movement detection, consist of two layered networks, called the primary visual cortex (V1), followed by the middle temporal area (MT), in which nonlinear functions will play important roles in the visual systems. These networks will be decomposed to asymmetric sub-networks with nonlinearities. In this paper, the fundamental characteristics in asymmetric neural networks with nonlinearities, are discussed for the detection of the changing stimulus or the movement detection in these neural networks. By the optimization of the asymmetric networks, movement detection equations are derived. Then, it was clarified that the even-odd nonlinearity combined asymmetric networks, has the ability in the stimulus change detection and the direction of movement or stimulus, while symmetric networks need the time memory to have the same ability. These facts are applied to two layered networks, V1 and MT.

## 1 Introduction

Visual information is processed firstly in the retinal network. Horizontal and bipolar cell responses are linearly related to the input modulation of stimulus light, while amacrine cells work linearly and nonlinearly in their responses [3], [10], [12]. Naka et al presented a simplified, but essential network of catfish inner retina [9]. Among prominent features of the visual networks, movement detections are carried out in the visual cortex. The visual cortex for the movement detection, consist of two layered networks, called the primary visual cortex (V1), followed by the middle temporal area (MT). The computational model of networks in visual cortex V1 and MT, was developed by Simoncelli and Heeger [16]. The model networks in V1 and MT, are identical in their structure. The computation is performed in two stages of the identical architecture, corresponding to networks in cortical areas V1 and MT.

In this paper, first, we analyze the asymmetric network with second order nonlinearity, which is based on the catfish retina for the detection of the changing and movement stimulus. We show the asymmetric network has the powerful ability in the detection of the changing stimulus and movement. Next, we present the network model developed by Simoncelli and Heeger [16], which shows first a linear receptive field, followed by half-squaring rectification and normalization in V1 and next V1 afferents, followed by half-squaring rectification and normalization. The half-squaring nonlinearity and normalization is analyzed by the approximation of Taylor series. Thus, the model is transformed in the parallel network structures, decomposed into asymmetric sub-networks.

## 2 Asymmetric Neural Network in the Retina

First, we present the asymmetric neural network in the catfish retina, which was studied by Naka, et al [3, 7, 9, 12] as shown in Fig. 1. A biological network of catfish retina shown in Fig. 1, might process the spatial interactive information between bipolar cells  $B_1$  and  $B_2$ . The bipolar  $B$  cell response is linearly related to the input modulation of light. The  $C$  cell shows an amacrine cell, which plays an important roll in the nonlinear function as squaring of the output of the bipolar cell  $B_2$ .

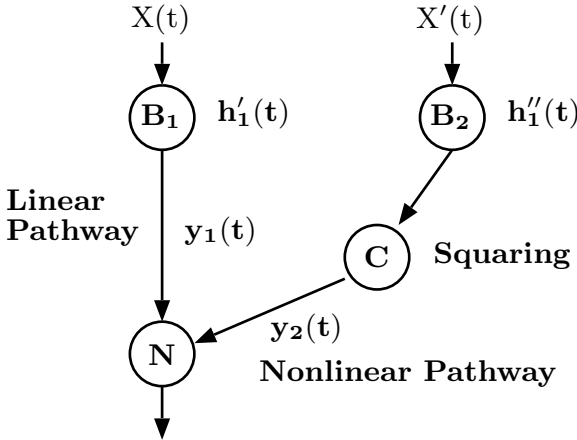
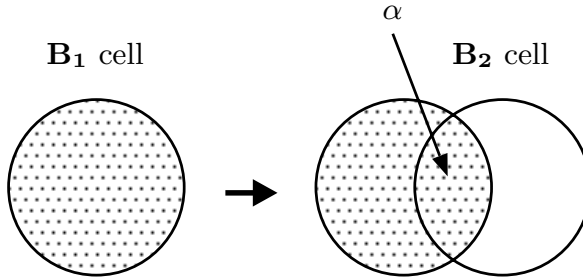


Fig. 1. Asymmetric neural network with linear and squaring nonlinear pathways

The  $N$  amacrine cell was clarified to be time-varying and differential with band-pass characteristics in the function. It is shown that  $N$  cell response is realized by a linear filter, which is composed of a differentiation filter followed by a low-pass filter. Thus the asymmetric network in Fig. 3 is composed of a linear pathway and a nonlinear pathway.

### 3 Asymmetric Neural Network with Quadratic Nonlinearity

Stimulus and movement perception are carried out firstly in the retinal neural network. Asymmetric neural network in the catfish retina, has characteristic asymmetric structure with a quadratic nonlinearity as shown in Fig. 1. Figure 2 shows a schematic diagram of changing stimulus or motion problem in front of the asymmetric network in Fig. 1.



**Fig. 2.** Schematic diagram of motion problem for spatial interaction

The slashed light is assumed to move from the left side to the right side, gradually. For the simplification of the analysis of the spatial interaction, we assume here the input functions  $x(t)$  and  $x''(t)$  to be Gaussian white noise, whose mean values are zero, but their deviations are different in their values. In Fig. 2, moving stimulus shows that  $x(t)$  merges into  $x''(t)$ , thus  $x''(t)$  is mixed with  $x(t)$ . Then, we indicate the right stimulus by  $x'(t)$ . By introducing a mixed ratio,  $\alpha$ , the input function of the right stimulus, is described in the following equation, where  $0 \leq \alpha \leq 1$  and  $\beta = 1 - \alpha$  hold. Then, Fig. 2 shows that the moving stimulus is described in the following equation,

$$x'(t) = \alpha x(t) + \beta x''(t) . \quad (1)$$

Let the power spectrums of  $x(t)$  and  $x''(t)$ , be  $p$  and  $p'$ , respectively and an equation  $p = kp''$  holds for the coefficient  $k$ , because we assumed here that the deviations of the input functions are different in their values. Figure 2 shows that the slashed light is moving from the receptive field of  $\mathbf{B}_1$  cell to the field of the  $\mathbf{B}_2$  cell. The mixed ratio of the input  $x(t)$ ,  $\alpha$  is shown in the receptive field of  $\mathbf{B}_2$  cell. First, on the linear pathway of the asymmetrical network in Fig. 1, the input function is  $x(t)$  and the output function is  $y(t)$ , which is an output after the linear filter of cell  $\mathbf{N}$ .

$$y(t) = \int h_1'''(\tau)(y_1(t - \tau) + y_2(t - \tau))d\tau + \varepsilon, \quad (2)$$

where  $y_1(t)$  shows the linear information on the linear pathway,  $y_2(t)$  shows the nonlinear information on the nonlinear pathway and  $\varepsilon$  shows error value.



The  $y_1(t)$  and  $y_2(t)$  are given, respectively as follows,

$$y_1(t) = \int_0^\infty h_1'(\tau)x(t-\tau)d\tau \quad (3)$$

$$y_2(t) = \int_0^\infty \int_0^\infty h_1''(\tau_1)h_1''(\tau_2)x'(t-\tau_1)x'(t-\tau_2)d\tau_1d\tau_2 . \quad (4)$$

We assume here the linear filter  $\mathbf{N}$  to have only summation operation without in the analysis. Thus the impulse response function  $h_1'''(t)$  is assumed to be value 1 without loss of generality.

### 3.1 Optimization of Asymmetric Neural Network

Under the assumption that the impulse response functions,  $h_1'(t)$  of the cell  $\mathbf{B}_1$ ,  $h_1''(t)$  of the cell  $\mathbf{B}_2$  and moving stimulus ratio  $\alpha$  in the right to be unknown, the optimization of the network is carried out. By the minimization of the mean squared value  $\xi$  of  $\varepsilon$  in (2), the following necessary equations for the optimization of (5) and (6), are derived,

$$\frac{\partial \xi}{\partial h_1'(t)} = 0, \quad \frac{\partial \xi}{\partial h_1''(t)} = 0 \quad \text{and} \quad \frac{\partial \xi}{\partial \alpha} = 0 . \quad (5)$$

Then, the following three equations are derived from the conditions for the optimization of (5)

$$\begin{aligned} E[y(t)x(t-\lambda)] &= h_1'(\lambda)p \\ E[(y(t) - C_0)x(t-\lambda_1)x(t-\lambda_2)] &= 2p^2\alpha^2 h_1''(\lambda_1)h_1''(\lambda_2) \\ E[(y(t) - C_0)x'(t-\lambda_1)x'(t-\lambda_2)] &= 2p'^2 h_1''(\lambda_1)h_1''(\lambda_2), \end{aligned} \quad (6)$$

where  $C_0$  is the mean value of  $y(t)$ , which is shown in the following. Here, (6) can be rewritten by applying Wiener kernels, which are related with input and out put correlations by Lee and Schetzen [19]. First, we can compute the 0-th order Wiener kernel  $C_0$ , the 1-st order one  $C_{11}(\lambda)$ , and the 2-nd order one  $C_{21}(\lambda_1, \lambda_2)$  on the linear pathway by the cross-correlations between  $x(t)$  and  $y(t)$ . The suffix  $i, j$  of the kernel  $C_{ij}(\cdot)$ , shows that  $i$  is the order of the kernel and  $j = 1$  means the linear pathway, while  $j = 2$  means the nonlinear pathway. Then, the 0-th order kernel under the condition of the spatial interaction of cell's impulse response functions  $h_1'(t)$  and  $h_1''(t)$ , are derived as follows.

The 1-st order kernel is defined by the correlation between  $y(t)$  and  $x(t)$  as on the linear pathway, and the correlation value is derived from the optimization condition of (6). Then, the following 1-st order kernel is shown

$$C_{11}(\lambda) = \frac{1}{p}E[y(t)x(t-\lambda)] = h_1'(\lambda) . \quad (7)$$

The 2-nd order kernel is defined and it becomes from the optimization condition of (6), in the following,

$$C_{21}(\lambda_1, \lambda_2) = \alpha^2 h_1''(\lambda_1)h_1''(\lambda_2) . \quad (8)$$

From (II), (VII) and (VIII), the ratio  $\alpha$  is a mixed coefficient of  $x(t)$  to  $x'(t)$ , is shown by  $\alpha^2$  as the amplitude of the second order Wiener kernel. On the nonlinear pathway, the 1-st order kernel  $C_{12}(\lambda)$  and the 2-nd order kernel  $C_{22}(\lambda_1, \lambda_2)$  is defined by the cross-correlations between  $x(t)$  and  $y(t)$  as shown in the following. Here,  $C_{22}(\lambda_1, \lambda_2)$  is computed from the optimization condition of (V), while  $C_{12}(\lambda)$  is derived as an additional condition from  $C_{11}(\lambda)$

$$C_{12}(\lambda) = \frac{\alpha}{\alpha^2 + k(1 - \alpha)^2} h_1'(\lambda) \quad (9)$$

and

$$C_{22}(\lambda_1, \lambda_2) = h_1''(\lambda_1) h_1''(\lambda_2) . \quad (10)$$

The motion problem is how to detect the movement in the increase of the ratio  $\alpha$  in Fig. 2. This implies that for the motion of the light from the left side circle to the right one, the ratio  $\alpha$  can be derived from the kernels described in the above, in which the second order kernels  $C_{21}$  and  $C_{22}$  are abbreviated in the representation of (VIII) and (X),

$$(C_{21}/C_{22}) = \alpha^2 \quad (11)$$

holds. Then, from (XI) the ratio  $\alpha$  is shown as follows

$$\alpha = \sqrt{\frac{C_{21}}{C_{22}}} . \quad (12)$$

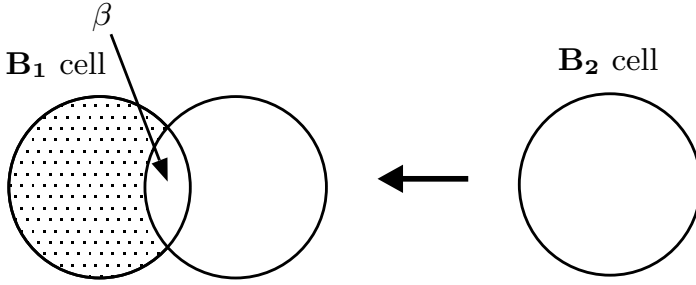
Equation (12) is called here  $\alpha$ -equation, which implies the change of the movement stimulus on the network and shows the detection of the movement change by the  $\alpha$ . Equation (12) does not show the direction of the movement. Any measure to show the direction of the movement, is needed. We will discuss how to detect the direction of the movement.

From the first order kernels  $C_{11}$  and  $C_{12}$ , and the second order kernels which are abbreviated in the time function, the following movement equation is derived

$$\frac{C_{12}}{C_{11}} = \frac{\sqrt{\frac{C_{21}}{C_{22}}}}{\frac{C_{21}}{C_{22}} + k \left( 1 - \sqrt{\frac{C_{21}}{C_{22}}} \right)^2} , \quad (13)$$

where  $k$  in the equation shows the difference of the power of the stimulus between the receptive fields of the left and the right cells  $\mathbf{B}_1$  and  $\mathbf{B}_2$ . Here, we have a problem whether (13) is different from the equation derived, under the condition that the movement of the stimulus from the right to the left. If (13) from the left to the right, is different from the equation from the right to the left, these equations will suggest different movement directions, as the vectors.

In the opposite direction from the right to left side stimulus, the schematic diagram of the stimulus movement, is shown in Fig. 3.



**Fig. 3.** Schematic diagram of the stimulus movement from right to left

$$C_{11}(\lambda) = h_1'(\lambda) \quad (14)$$

$$C_{21}(\lambda_1, \lambda_2) = \frac{k^2 \beta^2}{(\alpha^2 + k\beta^2)^2} h_1''(\lambda_1) h_1''(\lambda_2) . \quad (15)$$

Similarly, the following equations are derived on the nonlinear pathway,

$$\begin{aligned} C_{12}(\lambda) &= \beta h_1'(\lambda) \\ C_{22}(\lambda_1, \lambda_2) &= h_1''(\lambda_1) h_1''(\lambda_2) . \end{aligned} \quad (16)$$

From (14) and (16), the ratio  $\beta$  is derived, which is abbreviated in the notation

$$\beta = \frac{C_{12}}{C_{11}} \quad (17)$$

and the following equation is derived

$$\frac{C_{11}}{C_{12}} = \frac{k \sqrt{\frac{C_{21}}{C_{22}}}}{\left(1 - \frac{C_{21}}{C_{11}}\right)^2 + k \left(\frac{C_{12}}{C_{11}}\right)^2} . \quad (18)$$

It is important to show that (13) and (18) are different, that is, (13) implies the stimulus movement from left to right, while (18) implies the stimulus movement from right to left. We prove this proposition in the following.

Let  $(C_{12}/C_{11}) = X$  and  $\sqrt{C_{21}/C_{22}} = Y$  be set on (13) and (18). Then, equations are described as follows, respectively

$$Y = \frac{kX}{(1-X)^2 + kX^2} \quad \text{and} \quad X = \frac{Y}{Y^2 + k(1-Y)^2} .$$

By combining the above equations described in  $X$  and  $Y$ , the following equation holds,

$$\{(1-X)^2 + kX^2\}\{(X-1)^2 + k(X^2 - 2X - 1)\} + 2k^2X^2 = 0 . \quad (19)$$

Equation holds only in case of  $X = 1$ , that is  $\beta = 1$  from (17). This case implies the end of the stimulus movement in (18) and (13). During the stimulus movement, (19) shows positive values. Thus, (13) shows the directional movement from left to right according to the increase of  $\alpha$ , while (18) shows the directional movement from left to right according to the increase of  $\beta$ .

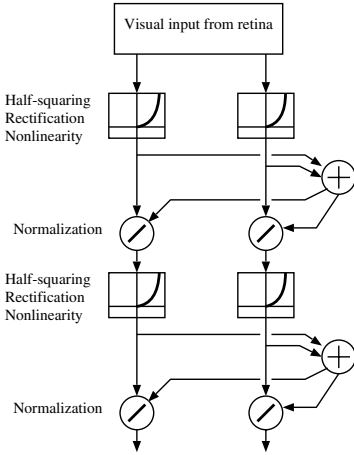
We call here an asymmetric network with odd-even nonlinearities in case of Fig. 1. Here we have questions, what is the behavior of another asymmetric networks with odd-odd nonlinearities or even-even nonlinearities. It is shown that the asymmetric network with the odd (even) nonlinearity on the left and another odd (even) nonlinearity on the right, cannot detect the direction of movement or stimulus. Thus, the asymmetric network with the combination of the odd and the even nonlinearities, has the powerful ability to detect both the change and the direction of the movement or stimulus, while symmetric networks need the time memory to have the same ability.

## 4 Layered Cortex Network by Asymmetric Networks

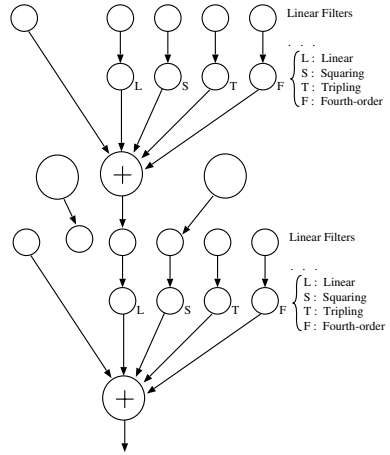
Based on the above analysis in the asymmetric nonlinear networks, the parallel processing in the brain cortex network, is discussed as follows. The visual cortex areas, V1 and MT, are studied to have the role of the tracking of the moving stimulus [14], [15], [16]. The nonlinearity in the visual system, has a problem of the it's order [13]. Heeger and Simoncelli presented a parallelization network model with half-wave squaring rectification nonlinearity in V1 and MT cortex areas in the brain [14], [15], [16]. In this paper, this parallel processing model is interpreted analytically to have the tracking function of the moving stimulus, which is based on the asymmetrical nonlinear networks described in the above.

Figure 4 shows a layered network model of cortex areas V1 followed by MT, which is suggested by Heeger and Simoncelli [16]. The nonlinear operation of the half-squaring nonlinearity and the normalization (saturation), is approximated by a sigmoid function. Then, the sigmoid function is represented by Taylor series, which includes the 1-st, the 2-nd, the 3-rd, the 4-th ... and higher orders nonlinearity terms. Thus, the network model in Fig. 4, is transformed to that of layered decomposed asymmetric model as shown in Fig. 5. The function of the layered decomposed networks in Fig. 5, is based on the asymmetric network with nonlinearities in Sect. 3. In V1 decomposed network in Fig. 5, the left side linear pathway (1-st order nonlinearity) and the right side nonlinear pathway, computes the  $\alpha$ -equation (detection of stimulus) and the movement equation (detection of the moving direction). When we pick up two parallel pathways with half-wave rectification, the combination of the odd order nonlinearity on the left side pathway and the even order nonlinearity on the right side pathway, can detect the movement and the direction of the stimulus and vice versa in the nonlinearities on the pathways.

The combination of the 1-st order on the left pathway and the even order nonlinearities (2-nd, 4-th, 6-th ... orders) on the right pathway in the first layer V1 in Fig. 5, has both abilities of detection of stimulus and the direction of



**Fig. 4.** Layered model of V1 followed by MT



**Fig. 5.** Layered decomposed networks of Fig. 4

stimulus, while that of the 1-st order left pathway and the odd order nonlinearities (1-st, 3-rd, 5-th . . . orders) in the right pathway in the first layer V1 in Fig. 4, can not detect the direction of the stimulus. This shows that only the first layer V1, is weak in the detection of the stimulus compared to the second layer MT, since these odd order nonlinearities in V1, has transformed to the even order nonlinearities (2-nd, 6-th, 10-th . . . orders) in the second layer MT in Fig. 5. This shows that the second layer MT, has both strong abilities of the detection of the stimulus and the direction of the stimulus. It is proved that the  $\alpha$ -equation and the movement equation are not derived in the symmetric network with the even order nonlinearity on the pathway and the other even nonlinearity on the another pathway. Both the even (or odd) nonlinearities on the parallel pathways, do not have strong works on the correlation processing. The even and the odd nonlinearities together, play an important role in the movement correlation processing.

## 5 Conclusion

It is important to study the biological neural networks from the view point of their structures and functions. In this paper, the neural networks in the biological retina and the cortical areas V1 and MT, are discussed to make clear the function of their stimulus changing or movement detection ability. Then, the networks are decomposed to the asymmetric networks with nonlinear higher order terms. The asymmetric network is fundamental in the biological neural network of the catfish retina. First, it is shown that the asymmetric networks with nonlinear quadratic characteristic, has the ability to detect moving stimulus. Then, the ability is described by the moving detection equation ( $\alpha$ -equation) and the movement direction equation. It was clarified that the asymmetric network with the even

and the odd nonlinearities, has efficient ability in the movement detection. Based on the asymmetric network principle, it is shown that the parallel-symmetric network with half-wave rectification of V1 and MT, has efficient ability in the movement detection.

## References

1. Hassenstein, B. and Reichard, W., "Systemtheoretische analyse der zeit-, reihenfolgen- and vorzeichenbewertung bei der bewegungsperzeption des rasselkäfers", *Chlorophanus. Z. Naturf.*, 11b, pp.513–524, 1956.
2. Barlow, H.B. and Levick, R.W., "The mechanism of directional selectivity in the rabbit's retina", *J. Physiol.* 173: pp.377–407, 1965.
3. Victor J.D. and Shapley K.M., "The nonlinear pathway of Y ganglion cells in the cat retina", *J. Gen. Physiol.*, vol.74, pp.671–689, 1979.
4. Ishii, N. and Naka, K.-I., "Movement and Memory Function in Biological Neural Networks", *Int. J. of Artificial Intelligence Tools*, Vol.4, No.4, pp.489–500, 1995.
5. Ishii, N., Sugiura, S., Nakamura, M., Yamauchi, S., "Sensory Perception, Learning and Integration in Neural Networks", *Proc. IEEE Int. Conf. on Information Intelligence & Systems*, pp.72–79, 1999.
6. Ishii, N. and Naka, K.-I., "Function of Biological Asymmetrical Neural Networks", *Biological and Artificial Computation: From Neuroscience to Technology*, LNCS, vol.1240, Springer, pp.1115–1125, 1997.
7. Korenberg, M.J., Sakai, H.M. and Naka, K.-I., "Dissection of the neuron network in the catfish inner retina", *J. Neurophysiol.* 61: pp.1110–1120, 1989.
8. Marmarelis, P.Z. and Marmarelis, V.Z., *Analysis of Physiological System: The White Noise Approach*, New York: Plenum Press, 1978.
9. Naka, K.-I., Sakai, H.M. and Ishii, N., "Generation of transformation of second order nonlinearity in catfish retina", *Annals of Biomedical Engineering*, 16: pp.53–64, 1988.
10. Shapley, R., "Visual cortex: pushing the envelope", *Nature: neuroscience*, vol.1, pp.95–96, 1998.
11. Reichardt, W., *Autocorrelation, a principle for the evaluation of sensory information by the central nervous system*, Rosenblith Edition., Wiley, 1961.
12. Sakuranaga, M. and Naka, K.-I., "Signal transmission in the catfish retina. III. Transmission to type-C cell", *J. Neurophysiol.* 58: pp.411–428, 1987.
13. Taub, E., Victor, J.D., and Conte, M.M., "Nonlinear preprocessing in short-range motion", *Vision Research*, 37: pp.1459–1477, 1997.
14. Heeger, D.J., "Modeling simple-cell direction selectivity with normalized, half-squared, linear operators", *J. Neurophysiol.* 70 : pp.1885–1898, 1993.
15. Heeger, D.J., Simoncelli, E.P., and Movshon, J.A., "Computational models of cortical visual processing", *Proc. Natl. Acad. Sci. USA*, vol.93, pp.623–627, 1996.
16. Simoncelli E.P., and Heeger, D. J., "A Model of Neuronal Responses in Visual Area MT", *Vision Research*, vol.38, pp.743–761, 1998.
17. Lu, Z.L., and Sperling, G., "Three-systems theory of human visual motion perception: review and update", *J. Opt. Soc. Am. A*, Vol.18, pp.2331–2370, 2001.
18. Ishii, N., Deguchi, T., and Sasaki, H., "Parallel Processing for Movement Detection in Neural Networks with Nonlinear Functions", *Intelligent Data Engineering and Automated Learning*, LNCS, vol.3177, pp.626–633, 2004.
19. Lee, Y.W., and Schetzen, M., "Measurement of the Wiener kernels of a nonlinear system by cross-correlation", *Int. J. Control*, vol.2, pp.237–254, 1965.

# Properties of the Hermite Activation Functions in a Neural Approximation Scheme

Bartłomiej Beliczynski

Warsaw University of Technology,  
Koszykowa 75, 00-662 Warsaw, Poland  
B.Beliczynski@ee.pw.edu.pl

**Abstract.** The main advantage to use Hermite functions as activation functions is that they offer a chance to control high frequency components in the approximation scheme. We prove that each subsequent Hermite function extends frequency bandwidth of the approximator within limited range of well concentrated energy. By introducing a scaling parameter we may control that bandwidth.

## 1 Introduction

In the end of eighties and beginning of nineties of the last century, neural network universal approximation properties were established [2], [4], [6], [11]. Since that it was clear that one-hidden-layer neural architecture with an appropriate activation function could approximate any function from a class with any degree of accuracy, provided that number of hidden units was sufficiently large. It was usually assumed that every single hidden unit had the same activation function. The most general case was presented in [11], where activation functions were taken from a wide class of functions i.e. non-polynomial.

A problem with contemporary computer implementation of neural schemes is that the activation function is usually implemented as its expansion in Taylor series i.e. by means of polynomials. So it is exactly what one wants to avoid. The finite number of hidden units and the polynomial type of activation functions limit accuracy. More hidden units and higher degree of polynomials may improve achievable accuracy.

It is well known that a particular type of activation function can determine certain properties of the network. Long running discussions about superiority of sigmoidal networks over radial bases networks or vice versa are good examples of that. In various observations it was established that one can simplify the network, achieving also a good generalization properties if various types of units are used. But the way to choose the most appropriate types of units remained unknown. Only a special incremental architecture suggested in [3] was suitable to handle this problem. But in fact it was only a trial and error method applied in every incremental approximation step.

Recently in several publications (see for instance [12], [14]) it was suggested to use Hermite functions as activation functions, but for every hidden unit a

different function. By using this method in [12] very good properties of EKG signals classification was presented. In [14] so called "constructive" approximation scheme was used which is a type of incremental approximation developed in [9], [10].

In this paper we analyse Hermite functions properties from function approximation perspective. We prove that each subsequent Hermite function extends frequency bandwidth of the approximator within a limited range of well concentrated energy. By introducing a scaling parameter one may control this bandwidth influencing at the same time the input argument dynamic range. Apart from formal statements and proves the paper is also focused on qualitative judgement and interpretations.

This paper is organised as follows. In Sect. 2 and 3 respectively a short description of a regularised approach to function approximation and the Hermite functions concept are recalled. In Sect. 4, some properties of the Hermite functions important from approximation point of view are formulated and proved. The main result of the paper is there. Sect. 5 contains a simple but practically useful generalisation of the result of Sect. 4 . Finally conclusions are drawn in Sect. 6.

## 2 The Function Approximation Framework

One of very typical neural network problem is a function approximation problem, which could be stated as follows. Let an unknown function be represented by a finite set of pairs

$$S = \left\{ (x_i, y_i) \in \mathbb{R}^d \times \mathbb{R} \right\}_{i=1}^N .$$

Find such function  $f \in \mathcal{F}$ , where  $\mathcal{F}$  is a function space, which minimises the following functional  $J(f) = \epsilon(f, S) + \lambda\varphi(f)$  .

The component  $\epsilon(f, S)$  represents an error function calculated on  $S$ ,  $\varphi(f)$  is a stabilizer which reinforces smoothness of the solution. Its values do not depend on  $S$ . Quite often used stabilizer is

$$\varphi(f) = \int_{\mathbb{R}^d} ds \frac{|\tilde{f}(s)|^2}{\tilde{g}(s)} ,$$

where  $\tilde{f}$  denotes Fourier transform of  $f$ ,  $\tilde{g}$  is a positive function and

$$\lim_{\|s\| \rightarrow \infty} \tilde{g}(s) = 0 .$$

Thus  $1/\tilde{g}$  is a high pass filter. The stabilizer reinforces that in the minimisation process,  $\tilde{f}(s)$  become a low pass filter ensuring that the stabilizer integral converges. In fact this function approximation formulation ensures that in a solution of the problem the high frequency components are well balanced by the parameter  $\lambda$  with error of approximation on a given data set. There exist a solution to the regularised problem see for instance [5] and many special interpretations see for example [8].



The property of controlling high frequency components in the approximation scheme may be achieved in the following linear combinations of the Hermite functions

$$f_n(x) = \sum_{i=0}^n w_i e_i(\text{lin}(x, a_i)) \quad (1)$$

where  $e_i$  denotes  $i$ -th Hermite function,  $\text{lin}(x, a_i) = a_{i0} + \sum_{k=1}^d a_{ik} x_k$ ,  $i = 0, \dots, n$  and  $a_i \in \mathbb{R}^{d+1}$  is a  $d+1$  element vector  $a_i = [a_{i0}, a_{i1}, \dots, a_{id}]^T$  and  $x_k \in \mathbb{R}$ ,  $k = 1, \dots, d$ . Each of  $e_i$  function is characterised by an easily determined limited bandwidth of well concentrated energy of function, what will be demonstrated in the rest of the paper.

### 3 Hermite Functions

In this section several concepts related to Hermite polynomials and Hermite functions will be recalled.

Let consider a space of great practical interest  $L^2(-\infty, +\infty)$  with the inner product defined  $\langle x, y \rangle = \int_{-\infty}^{+\infty} x(t)y(t)dt$ . In such space a sequence of linearly independent functions could be created as follows  $1, t, \dots, t^n, \dots$ . Another sequence ensuring that every element of it is bounded is  $h_0(t) = w(t) = e^{-t^2/2}$ ,  $h_1(t) = tw(t), \dots, h_n(t) = t^n w(t), \dots$

The last sequence forms an useful basis for function approximation in the space  $L^2(-\infty, +\infty)$ . This basis could be orthonormalised by using well known and efficient Gram-Schmidt process (see for instance [7]). Finally one obtains a new, now orthonormal basis spanning the same space

$$e_0(t), e_1(t), \dots, e_n(t), \dots \quad (2)$$

where

$$e_n(t) = c_n e^{-\frac{t^2}{2}} H_n(t); \quad H_n(t) = (-1)^n e^{t^2} \frac{d^n}{dt^n} (e^{-t^2}); \quad c_n = \frac{1}{(2^n n! \sqrt{\pi})^{1/2}} \quad (3)$$

The polynomials  $H_n(t)$  are called Hermite polynomials and the functions  $e_n(t)$  Hermite functions. Some standard mathematical properties of the Hermite polynomials are listed in the Appendix.

### 4 Properties of the Hermite Activation Functions

According to (3) the first several Hermite functions could be calculated

$$\begin{aligned} e_0(t) &= \frac{1}{\pi^{1/4}} e^{-\frac{t^2}{2}}, & e_1(t) &= \frac{1}{\sqrt{2\pi^{1/4}}} e^{-\frac{t^2}{2}} 2t, \\ e_2(t) &= \frac{1}{2\sqrt{2\pi^{1/4}}} e^{-\frac{t^2}{2}} (4t^2 - 2), & e_3(t) &= \frac{1}{4\sqrt{3\pi^{1/4}}} e^{-\frac{t^2}{2}} (8t^3 - 12t). \end{aligned}$$

One can determine next  $e_n(t)$  functions by using recursive formulae. Its certain properties further useful are listed in Proposition [1](#)

**Proposition 1.** *The following hold*

$$e_{n+1}(t) = \sqrt{\frac{2}{n+1}}te_n(t) - \sqrt{\frac{n}{n+1}}e_{n-1}(t); \quad n = 1, 2, 3, \dots \quad (4)$$

$$\frac{d}{dt}e_n(t) = -te_n(t) + \sqrt{2n}e_{n-1}(t); \quad n = 1, 2, 3, \dots \quad (5)$$

$$\frac{d^2}{dt^2}e_n(t) = e_n(t)(t^2 - (2n + 1)); \quad n = 0, 1, 2, 3, \dots \quad (6)$$

*Proof.* Simply from [\(3\)](#)  $e_{n+1}(t) = c_{n+1}e^{-\frac{t^2}{2}}H_{n+1}(t)$ . Using formulae of the Appendix and a simple algebraic manipulations one obtains

$$e_{n+1}(t) = t\left(2\frac{c_{n+1}}{c_n}\right)c_n e^{-\frac{t^2}{2}}H_n(t) - n\left(2\frac{c_{n+1}}{c_n}\right)c_{n-1}e^{-\frac{t^2}{2}}H_{n-1}(t)$$

and finally [\(4\)](#). The equation [\(5\)](#) proof is a straightforward and the derivation of [\(6\)](#) requires of using properties listed in the Appendix.

In quantum mechanics [\(6\)](#) is known as a harmonic oscillator (Schrodinger equation).

*Remark 1.* One can noticed from [\(6\)](#) that the only inflection points which are not zeros of the Hermite functions are located at  $\pm t_n, t_n = \sqrt{2n + 1}$ .

**Proposition 2.** *For every  $t \geq t_n = \sqrt{2n + 1}$ ,  $e_n(t) > 0$  and  $\lim_{t \rightarrow \pm\infty} e_n(t) = 0$ .*

The range  $[-\sqrt{2n + 1}, \sqrt{2n + 1}]$  can be treated as an admissible input argument dynamic range. Outside of this range  $e_n(t)$  is smoothly decaying.

Several functions of the Hermite basis are shown in Fig. [1](#). There are marked there positive inflection points which are not zeros of the function.

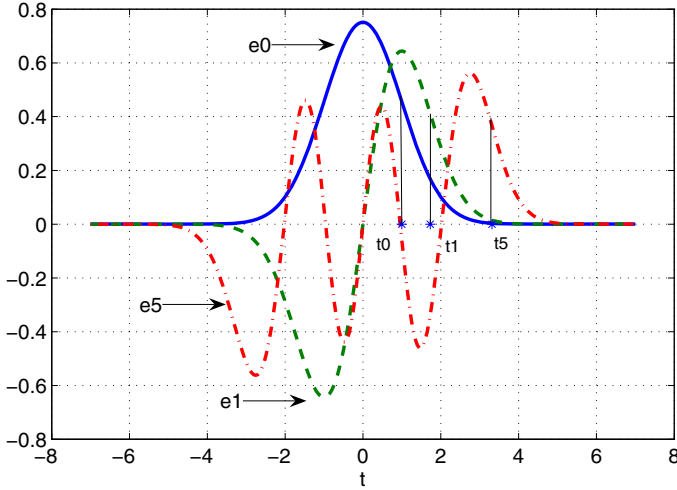
It is well known that Hermite functions are eigenfunctions of the Fourier transform. However we will put that fact into Proposition [3](#) and simply prove it in few lines. In many publications (see for instance [11](#)) the eigenvalue of this transformation is different than the one presented here.

Let  $\tilde{e}_n(j\omega)$  denotes a Fourier transform of  $e_n(t)$  defined as follows

$$\tilde{e}_n(j\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e_n(t)e^{-j\omega t} dt .$$

**Proposition 3.** *Let  $e_n(t)$  be as in [\(3\)](#), then*

$$\tilde{e}_n(j\omega) = e^{-j\frac{\pi}{2}n}e_n(\omega) \quad (7)$$



**Fig. 1.** Several Hermite functions with marked positive inflection points  $t_n$

*Proof.* For  $n = 0$  and  $n = 1$  one can calculate  $\tilde{e}_0(j\omega)$  and  $\tilde{e}_1(j\omega)$  obtaining  $\frac{1}{\pi^{1/4}}e^{-\frac{\omega^2}{2}}$  and  $-j\frac{1}{\sqrt{2\pi^{1/4}}}2\omega e^{-\frac{\omega^2}{2}}$  what satisfy (7). Let assume that (7) is fulfilled for  $n = k - 1$  and  $n = k$ . Now calculating Fourier transform of the formula (4) one obtains

$$\begin{aligned}
 \tilde{e}_{k+1}(j\omega) &= \sqrt{\frac{2}{k+1}} \left( -\frac{1}{j} \right) \frac{d}{d\omega} (\tilde{e}_k(j\omega)) - \sqrt{\frac{k}{k+1}} \tilde{e}_{k-1}(j\omega) \\
 &= (-j)^{k+1} \left( -\sqrt{\frac{2}{k+1}} \frac{d}{d\omega} (e_k(\omega)) + \sqrt{\frac{k}{k+1}} e_{k-1}(\omega) \right) \\
 &= (-j)^{k+1} \left( -\sqrt{\frac{2}{k+1}} (-\omega e_k(\omega) + \sqrt{2k} e_{k-1}(\omega)) + \sqrt{\frac{k}{k+1}} e_{k-1}(\omega) \right) \\
 &= (-j)^{k+1} \left( \omega \sqrt{\frac{2}{k+1}} e_k(\omega) - \sqrt{\frac{k}{k+1}} e_{k-1}(\omega) \right)
 \end{aligned}$$

and taking (4)

$$\tilde{e}_{k+1}(j\omega) = (-j)^{k+1} e_{k+1}(\omega) = e^{-j\frac{\pi}{2}(k+1)} e_{k+1}(\omega) .$$

The eigenvalue of Fourier transform of  $e_n$  is  $e^{-j\frac{\pi}{2}n}$ . In engineering interpretation one would say that the magnitude of a Hermite function and its Fourier transform are the same. This transform however introduces a constant phase shift equal to multiplicity of  $\frac{\pi}{2}$ . This feature of Hermite functions being eigenfunctions of Fourier transform is exploited in [13].

So  $|\tilde{e}_n(j\omega)|$  is the same as  $|e_n(t)|$  and the spectrum of magnitude  $|e_n(\omega)|$  will have inflection points at  $\pm\omega_n$ ,  $\omega_n = \sqrt{2n+1}$ . Those inflection points could be treated as markers for frequency bandwidth definition.

**Lemma 1.** (technical) Let  $I_n(\nu) = \int_{-\nu}^{\nu} e_n^2(\omega) d\omega$  then

$$I_n(\nu) = I_{n-1}(\nu) - \sqrt{\frac{2}{n}} e_{n-1}(\nu) e_n(\nu) .$$

*Proof.* Requires only simple but lengthily algebraic manipulations using results of Proposition [1](#) and is omitted here.

By a straightforward calculation we come to  $I_n(\nu) = I_0(\nu) - \sum_{i=1}^n \sqrt{\frac{2}{i}} e_{i-1}(\nu) e_i(\nu)$ .

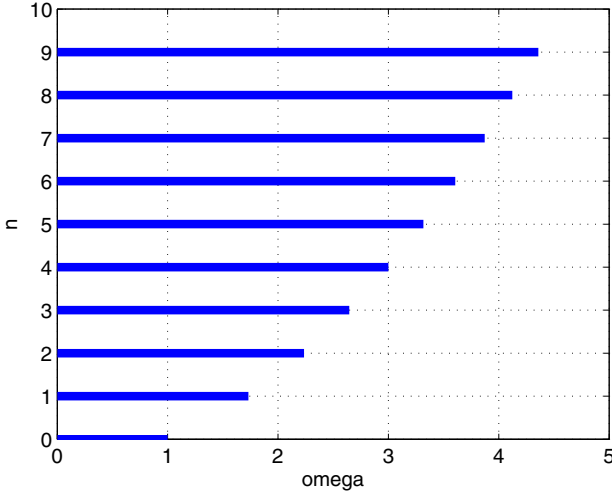
**Proposition 4.** The following hold:

1. if  $\nu_2 > \nu_1$  then  $I_n(\nu_2) > I_n(\nu_1)$
2. if  $\nu \geq \omega_n$  then  $I_n(\nu) < I_{n-1}(\nu)$
3.  $I_n(v + \Delta v) > I_{n-1}(v)$ ,  $\Delta v > 0$

*Proof.* ad 1) It comes directly from definition in Prop. [4](#); ad 2) if  $\nu \geq \omega_n$  then from Prop. [2](#)  $e_{n-1}(\nu) > 0$ ,  $e_n(\nu) > 0$  and the result is obtained from Prop. [1](#); ad 3) Let denote  $d(v) = I_n(v + \Delta v) - I_{n-1}(v)$ . By using Lemma [1](#) we calculate the integral change along the following path  $I_{n-1}(v) \rightarrow I_{n-1}(v + \Delta v) \rightarrow I_n(v + \Delta v)$ . It is  $d(v) = 2 \int_v^{v+\Delta v} e_{n-1}^2(\omega) d\omega - \sqrt{\frac{2}{n}} e_{n-1}(v) e_n(v)$ . Calculating  $\frac{d}{dv}(d(v))$  we obtain  $\frac{d}{dv}(d(v)) = 2e_{n-1}^2(v) - \sqrt{\frac{2}{n}}(-2ve_{n-1}(v)e_n(v) + \sqrt{2n}e_{n-1}^2(v) + \sqrt{2(n-1)}e_{n-2}(v)e_n(v)) = e_{n-1}^2(v) + 2\sqrt{\frac{2}{n}}e_{n-1}(v)e_n(v) - 2\sqrt{\frac{n-1}{n}}e_{n-2}(v)e_n(v)$ . Now using [4](#)  $\sqrt{\frac{n-1}{n}}e_{n-2}(v)e_n(v) = \sqrt{\frac{2}{n}}ve_{n-1}(v)e_n(v) - e_n^2(v)$ , and after simple manipulations one obtains  $\frac{d}{dv}(d(v)) = e_{n-1}^2(v) + 2e_n^2(v) > 0$ , so  $d(v)$  is an increasing function.

*Remark 2.* From 3. of Proposition [4](#) it is clear that  $I_n(\omega_n) > I_{n-1}(\omega_{n-1})$ .

By using Lemma [1](#), one can calculate  $I_n(\omega_n)$  for various  $n$  starting from  $I_0(\omega_0) = \frac{1}{\sqrt{\pi}} \int_{-1}^1 e^{-\omega^2} d\omega = \frac{2}{\sqrt{\pi}} \int_0^1 e^{-\omega^2} d\omega = \text{erf}(1) = 0.843$ ,  $I_1(\omega_1) = 0.889$ ,  $I_2(\omega_2) = 0.907$ ,  $I_3(\omega_3) = 0.915$  and  $I_9(\omega_9) = 0.938$ . One might say that for the Hermite function  $e_n$  at least 84,3% of its energy is concentrated within the frequency range  $\omega \in [-\sqrt{2n+1}, \sqrt{2n+1}]$ . For larger  $n$  even more function energy is concentrated in there. **The range  $[-\sqrt{2n+1}, \sqrt{2n+1}]$  we name the approximate bandwidth of  $e_n$ .** In Fig. [2](#) only positive parts of the approximate bandwidths are shown.



**Fig. 2.** The approximate bandwidths of  $e_n(\omega)$  for various  $n$

## 5 Scaling Parameter and the Bandwidth

From engineering point of view an important generalisation of the basis (2) comes when scaling of  $t$  variable by using a  $\sigma \in (0, \infty)$  parameter. So if one substitutes  $t := \frac{t}{\sigma}$  into (3) and modifies  $c_n$  to ensure orthonormality, then

$$e_n(t, \sigma) = c_{n,\sigma} e^{-\frac{t^2}{2\sigma^2}} H_n\left(\frac{t}{\sigma}\right) \quad \text{where } c_{n,\sigma} = \frac{1}{(\sigma 2^n n! \sqrt{\pi})^{1/2}} \quad (8)$$

and

$$e_n(t, \sigma) = \frac{1}{\sqrt{\sigma}} e_n\left(\frac{t}{\sigma}\right) \quad \text{and } \tilde{e}_n(\omega, \sigma) = \sqrt{\sigma} \tilde{e}_n(\sigma\omega) \quad (9)$$

Thus by introducing scaling parameter  $\sigma$  into (8) one may adjust the input argument dynamic range of  $e_n(t, \sigma)$  and its frequency bandwidth

$$t \in [-\sigma\sqrt{2n+1}, \sigma\sqrt{2n+1}]; \quad \omega \in \left[-\frac{1}{\sigma}\sqrt{2n+1}, \frac{1}{\sigma}\sqrt{2n+1}\right]$$

Moreover it is possible of course to select different scaling parameters for each Hermite function like  $e_n(t, \sigma_n)$  but in such a case the set of such functions is not orthonormal any more.

## 6 Conclusions

Despite the fact that Hermite functions are orthogonal, in the considered approximation scheme of the multivariable function approximation, the basis is

not orthogonal. The main advantages however to use Hermite functions is that they offer a chance to control high frequency components in the approximation scheme. Every  $e_i(t)$ ,  $i = 1, \dots, n$  is well bounded and for large  $t$  approaches zero. We proved that each subsequent Hermite function extends frequency bandwidth of the approximator within limited range of well concentrated energy. By introducing scaling parameters one may control the input dynamic range and at the same time frequency bandwidth of the Hermite functions.

For a traditional fixed activation function architecture, if one selects the same parameters for all hidden units, then the only one contributes to decrease of the approximation error. In the considered scheme in such a case all the units are still useful giving an orthonormal basis for approximation.

## References

1. Hermite polynomial. <http://mathworld.wolfram.com/HermitePolynomial.html>.
2. Cybenko, G.: Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, **2** (1989) 303–314
3. Fallman, S., Lebiere, C.: The cascade correlation learning architecture. Technical report, CMU-CS-90-100, (1991)
4. Funahashi, K.: On the approximate realization of continuous mappings by neural networks. *Neural Networks* **2** (1989) 183–192
5. Girosi, F., Jones, M., Poggio, T.: Regularization theory and neural networks architecture. *Neural Computation*, **7**(2) (1995) 219–269
6. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. *Neural Networks* **2** (1989) 359–366
7. Kreyszig, E.: *Introductory functional analysis with applications*. J.Wiley (1978)
8. Kurkova, V.: Supervised learning with generalisation as an inverse problem. *Logic Journal of IGPL*, **13** (2005) 551–559
9. Kwok, T., Yeung, D.: Constructive algorithms for structure learning in feedforward neural networks for regression problems. *IEEE Trans. Neural Netw.* **8**(3) (1997) 630–645
10. Kwok, T., Yeung, D.: Objective functions for training new hidden units in constructive neural networks. *IEEE Trans. Neural Networks* **8**(5) (1997) 1131–1148
11. Leshno, T., Lin, V., Pinkus, A., Schocken, S.: Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks* **13** (1993) 350–373
12. Linh, T. H.: *Modern Generations of Artificial Neural Networks and their Applications in Selected Classification Problems* (in Polish). Publishing House of the Warsaw University of Technology (2004)
13. Mackenzie, M. R., Tieu, A. K.: Hermite neural network correlation and application. *IEEE Trans. on Signal Processing* **51**(12) (2003) 3210–3219
14. Mora, I., Khorasani, K.: Constructive feedforward neural networks using hermite polynomial activation functions. *IEEE Transactions on Neural Networks* **16** (2005) 821–833

## Appendix

Definition of Hermite polynomials might be provided in various ways. We will adopt Rodrigues' formula

$$H_n(t) = (-1)^n e^{t^2} \frac{d^n}{dt^n} e^{-t^2} , \text{ where } n = 0, 1, 2, 3, \dots$$

Standard properties of the Hermite polynomials can be found in many mathematical textbooks or Web pages see for example [7], [1]. We restrict our attention to such formulas which are useful for this paper. These are the following.

$$H_n(-t) = (-1)^n H_n(t) ,$$

$$H_n(0) = \begin{cases} 0 & \text{for } n \text{ odd} \\ (-1)^{\frac{n}{2}} 2^{\frac{n}{2}} \prod_{i=1}^{n-1} i & \text{for } n \text{ even} \end{cases}$$

$$H_0(t) = 1 ,$$

$$H_{n+1}(t) = 2tH_n(t) - 2nH_{n-1}(t) ,$$

$$\frac{d}{dt} H_n(t) = 2nH_{n-1}(t) ,$$

$$\frac{d}{dt} (e^{-t^2} H_n(t)) = -e^{-t^2} H_{n+1}(t) ,$$

$$\int_0^x H_n(t) dt = \frac{H_{n+1}(x)}{2(n+1)} - \frac{H_{n+1}(0)}{2(n+1)} ,$$

$$\int_0^x e^{-t^2} H_n(t) dt = H_{n-1}(0) - e^{-x^2} H_{n-1}(x) ,$$

$$\frac{d^2}{dt^2} H_n(t) - 2t \frac{d}{dt} H_n(t) + 2nH_n(t) = 0 .$$

# Study of the Influence of Noise in the Values of a Median Associative Memory

Humberto Sossa, Ricardo Barrón, and Roberto A. Vázquez

Centro de Investigación en Computación-IPN  
Av. Juan de Dios Bátiz, esquina con Miguel Othón de Mendizábal  
Mexico City, 07738, Mexico  
hsossa@cic.ipn.mx, rbarron@cic.ipn.mx, ravem@ipn.mx

**Abstract.** In this paper we study how the performance of a median associative memory is influenced when the values of its elements are altered by noise. To our knowledge this kind of research has not been reported until know. We give formal conditions under which the memory is still able to correctly recall a pattern of the fundamental set of patterns either from a non-altered or a noisy version of it. Experiments are also given to show the efficiency of the proposal.

## 1 Introduction

An associative memory is a device designed to recall patterns. An associative memory (AM)  $\mathbf{M}$  can be viewed as an input-output system as follows:  $\mathbf{x} \rightarrow \mathbf{M} \rightarrow \mathbf{y}$ , with  $\mathbf{x}$  and  $\mathbf{y}$ , respectively the input and output patterns vectors. Each input vector forms an association with a corresponding output vector. The AM  $\mathbf{M}$  is represented by a matrix whose  $ij$ -th component is  $m_{ij}$ .  $\mathbf{M}$  is generated from a finite a priori set of known associations, known as the *fundamental set of associations*, or simply the *fundamental set* (FS).

If  $\xi$  is an index, the fundamental set is represented as:  $\{(\mathbf{x}^\xi = \mathbf{y}^\xi) \mid \xi \in \{1, 2, \dots, p\}\}$  with  $p$  the cardinality of the set. The patterns that form the fundamental set are called *fundamental patterns*. If it holds that  $\mathbf{x}^\xi = \mathbf{y}^\xi \forall \xi \in \{1, 2, \dots, p\}$ , then  $\mathbf{M}$  is auto-associative, otherwise it is hetero-associative. A distorted version of a pattern  $\mathbf{x}$  to be restored will be denoted as  $\tilde{\mathbf{x}}$ . If when feeding a distorted version of  $\mathbf{x}^w$  with  $w \in \{1, 2, \dots, p\}$  to an associative memory  $\mathbf{M}$ , then it happens that the output corresponds exactly to the associated pattern  $\mathbf{y}^w$ , we say that recalling is robust. If  $\mathbf{x}^w$  is not altered with noise, then recall is perfect. Several models for associative memories have emerged in the last 40 years. Refer for example to [1]-[10].

## 2 Basics of Median Associative Memories

Two associative memories are fully described in [5]. Due to space limitations, only hetero-associative memories are described. Auto-associative memories can be obtained simple by doing  $\mathbf{x}^\xi = \mathbf{y}^\xi \forall \xi \in \{1, 2, \dots, p\}$ . Let us designate hetero-associative median memories as HAM-memories. Let  $\mathbf{x} \in \mathbf{R}^n$  and  $\mathbf{y} \in \mathbf{R}^m$  two vectors.



To operate HAM memories two phases are required, one for memory construction and one for pattern recall.

## 2.1 Memory Construction

Two steps are required to build the HAM-memory:

**Step 1:** For each  $\xi=1,2,\dots,p$ , from each couple  $(\mathbf{x}^\xi, \mathbf{y}^\xi)$  build matrix:  $\mathbf{M}^\xi$  as:

$$\mathbf{M}^\xi = \begin{pmatrix} A(y_1, x_1) & A(y_1, x_2) & \cdots & A(y_1, x_n) \\ A(y_2, x_1) & A(y_2, x_2) & \cdots & A(y_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ A(y_m, x_1) & A(y_m, x_2) & \cdots & A(y_m, x_n) \end{pmatrix}_{m \times n}, \quad (1)$$

with  $A(x_i, y_j) = x_i - y_j$  as proposed in [4].

**Step 2:** Apply the median operator to the matrices obtained in Step 1 to get matrix  $\mathbf{M}$  as follows:

$$\mathbf{M} = \underset{\xi=1}{\text{med}} \left[ \mathbf{M}^\xi \right]. \quad (2)$$

The  $ij$ -th component  $\mathbf{M}$  is thus given as follows:

$$m_{ij} = \underset{\xi=1}{\text{med}} A(y_i^\xi, x_j^\xi). \quad (3)$$

## 2.2 Pattern Recall

We have two cases:

**Case 1: Recall of a fundamental pattern.** A pattern  $\mathbf{x}^w$ , with  $w \in \{1, 2, \dots, p\}$  is presented to the memory  $\mathbf{M}$  and the following operation is done:

$$\mathbf{M} \diamond_{\mathbf{B}} \mathbf{x}^w. \quad (4)$$

The result is a column vector of dimension  $n$ , with  $i$ -th component given as:

$$\left( \mathbf{M} \diamond_{\mathbf{B}} \mathbf{x}^w \right)_i = \underset{j=1}{\text{med}} B(m_{ij}, x_j^w), \quad (5)$$

with  $B(x_i, y_j) = x_i + y_j$  as proposed in [4].

**Case 2: Recall of a pattern from an altered version of it.** A pattern  $\tilde{\mathbf{x}}$  (altered version of a pattern  $\mathbf{x}^w$ ) is presented to the hetero-associative memory  $\mathbf{M}$  and the following operation is done:

$$\mathbf{M} \diamond_{\mathbf{B}} \tilde{\mathbf{x}}. \quad (6)$$

Again, the result is a column vector of dimension  $n$ , with  $i$ -th component given as:

$$\left( \mathbf{M} \diamond_{\mathbf{B}} \tilde{\mathbf{x}} \right)_i = \underset{j=1}{\text{med}} B(m_{ij}, \tilde{x}_j). \quad (7)$$

The following three propositions provide the conditions for perfect recall of a pattern of the FS or from an altered version of it. According to [5]:

**Theorem 1 [5].** Let  $\{(\mathbf{x}^\alpha = \mathbf{y}^\alpha) \mid \alpha = 1, 2, \dots, p\}$  with  $\mathbf{x}^\alpha \in \mathbf{R}^n$ ,  $\mathbf{y}^\alpha \in \mathbf{R}^m$  the fundamental set of an HAM-memory  $\mathbf{M}$  and let  $(\mathbf{x}^\gamma, \mathbf{y}^\gamma)$  an arbitrary fundamental couple with  $\gamma \in \{1, 2, \dots, p\}$ . If  $\text{med}_{j=1}^n \varepsilon_{ij} = 0$ ,  $i=1, \dots, m$ ,  $\varepsilon_{ij} = m_{ij} - A(y_i^\gamma, x_j^\gamma)$  then  $(\mathbf{M} \diamond_B \mathbf{x}^\gamma)_i = y_i^\gamma, i=1 \dots m$ .

**Corollary 1 [5].** Let  $\{(\mathbf{x}^\alpha = \mathbf{y}^\alpha) \mid \alpha = 1, 2, \dots, p\}$ ,  $\mathbf{x}^\alpha \in \mathbf{R}^n$ ,  $\mathbf{y}^\alpha \in \mathbf{R}^m$ . A HAM-median memory  $\mathbf{M}$  has correct recall if for all  $\alpha = 1, 2, \dots, p$ ,  $\mathbf{M}^\alpha = \mathbf{M}$  where  $\mathbf{M} = \mathbf{y}^\xi \diamond_A (\mathbf{x}^\xi)$  is the associated partial matrix to the fundamental couple  $(\mathbf{x}^\alpha, \mathbf{y}^\alpha)$  and  $p$  is the number of couples.

**Theorem 2 [5].** Let  $\{(\mathbf{x}^\alpha = \mathbf{y}^\alpha) \mid \alpha = 1, 2, \dots, p\}$  with  $\mathbf{x}^\alpha \in \mathbf{R}^n$ ,  $\mathbf{y}^\alpha \in \mathbf{R}^m$ , a FS with perfect recall. Let  $\eta^\alpha \in \mathbf{R}^n$  a pattern of mixed noise. A HAM-median memory  $\mathbf{M}$  has correct recall in the presence of mixed noise if this noise is of median zero, this is if  $\text{med}_{j=1}^n \eta_j^\alpha = 0, \forall \alpha$ .

In [7], the authors present new results concerning median associative memories.

### 2.3 Case of a General Fundamental Set

In [6] was shown that due to in general a fundamental set (FS) does not satisfy the restricted conditions imposed by Theorem 1 and its Corollary, in [6] it is proposed the following procedure to transform a general FS into an auxiliary FS' satisfying the desired conditions:

#### TRAINING PHASE:

**Step 1.** Transform the FS into an auxiliary fundamental set (FS') satisfying Theorem 1:

Make  $D$  a vector of constant values,  $D = [d \ d \ \dots \ d]^T$ ,  $d = \text{constant}$ .

Make  $(\mathbf{x}^1, \mathbf{y}^1) = (\mathbf{x}^1, \mathbf{y}^1)$ .

For the remaining couples do {

For  $\xi = 2$  to  $p$  {

$$\mathbf{x}^\xi = \mathbf{x}^{\xi-1} + D; \quad \hat{\mathbf{x}}^\xi = \mathbf{x}^\xi - \mathbf{x}^\xi; \quad \mathbf{y}^\xi = \mathbf{y}^{\xi-1} + D; \quad \hat{\mathbf{y}}^\xi = \mathbf{y}^\xi - \mathbf{y}^\xi \}}$$

**Step 2.** Build matrix  $\mathbf{M}$  in terms of set FS'; Apply to FS' steps 1 and 2 of the training procedure described at the beginning of this section.

#### RECALLING PHASE:

We have also two cases, i.e.:

**Case 1:** Recalling of a fundamental pattern of FS:

Transform  $\mathbf{x}^\xi$  to  $\hat{\mathbf{x}}^\xi$  by applying the following transformation:  $\hat{\mathbf{x}}^\xi = \mathbf{x}^\xi + \mathbf{x}^\xi$ .

Apply equations (4) and (5) to each  $\hat{\mathbf{x}}^\xi$  of FS' to recall  $\mathbf{y}^\xi$ .

Recall each  $\mathbf{y}^\xi$  by applying the following inverse transformation:  $\mathbf{y}^\xi = \hat{\mathbf{y}}^\xi - \mathbf{y}^\xi$ .

**Case 2:** Recalling of a pattern  $\mathbf{y}^\xi$  from an altered version of its key:  $\mathbf{x}^\xi$  :

Transform  $\mathbf{x}^\xi$  to  $\bar{\mathbf{x}}^\xi$  by applying the following transformation:  $\mathbf{x}^\xi = \bar{\mathbf{x}}^\xi + \hat{\mathbf{x}}^\xi$  .

Apply equations (6) and (7) to  $\bar{\mathbf{x}}^\xi$  to get  $\bar{\mathbf{y}}^\xi$  , and

Anti-transform  $\bar{\mathbf{y}}^\xi$  as  $\mathbf{y}^\xi = \bar{\mathbf{y}}^\xi - \hat{\mathbf{y}}^\xi$  to get  $\mathbf{y}^\xi$  .

### 3 Influence of Noise in Median Associative Memories

Until now all researchers have studied how noise added to patterns can affect the performance of an associative memory. To our knowledge, nobody has studied how the presence of noise in the values  $m_{ij}$  of a memory  $\mathbf{M}$ , not only in the pattern can influence the performance of the memory.

The study of the influence of the noise in the components of an associative memory is important for the following two reasons. In the one hand, the topology of an space of associative memories seen as a space of operators, and, in the other hand, the influence that the noise has in the recall of a fundamental pattern without and with noise.

In this section we formally study this for the case of the median associative memory (MEDIANMEM). We give a set of formal conditions under which an MEDIANMEM can still correctly recall a pattern of the FS either form unaltered or altered version of it.

The following proposition, whose proof is not given here due to space limitation, provides these conditions:

**Proposition 1.** *Let  $\{(\mathbf{x}^\alpha, \mathbf{y}^\alpha) \mid \alpha = 1, 2, \dots, p\}$ ,  $\mathbf{x}^\alpha \in \mathbf{R}^n$ ,  $\mathbf{y}^\alpha \in \mathbf{R}^m$  a FS with perfect recall. Let  $\eta \in \mathbf{R}^{mn}$  a pattern of mixed noise to be added to the memory. A HAM-median memory  $\mathbf{M}$  has perfect recall in the presence of mixed noise if this noise, per row, is of median zero, this is if  $\text{med}_{j=1}^n \eta_{i,j} = 0, i = 1, m$ .*

In general, noise added to a pattern does not satisfy the restricted conditions imposed by Proposition 1. The following proposition (in the transformed domain), whose proof is not given here due to space limitations, states the conditions under which MEDIANMEMS provide for perfect recall under general mixed noise:

**Proposition 2.** *Let  $\{(\mathbf{x}^\alpha, \mathbf{y}^\alpha) \mid \alpha = 1, 2, \dots, p\}$ ,  $\mathbf{x}^\alpha \in \mathbf{R}^n$ ,  $\mathbf{y}^\alpha \in \mathbf{R}^m$  a FS and  $\mathbf{M}$  its memory. Without lost of generality suppose that is  $p$  odd. Let  $M_i, i=1, \dots, m$  a row of matrix  $\mathbf{M}$  and  $\mathbf{x}$  a key pattern of the FS. If row  $M_i$  and pattern  $\mathbf{x}$  are both of size  $n$ , and the number of elements distorted by mixed noise from row  $M_i$  and  $\mathbf{x}$  is less than  $(n+1)/2-1$ , then  $\mathbf{y}^\xi = \tilde{\mathbf{M}} \diamond_B \bar{\mathbf{x}}$ .*

In other words, if when given a row of an associative memory, less than 50% of the elements of this row and the key pattern taken together are distorted by noise, then perfectly recall is always obtained.

Another useful proposition, whose proof is not given here due to space limitation, is also the following:

**Proposition 3.** Let  $\{(\mathbf{x}^\alpha, \mathbf{y}^\alpha) \mid \alpha = 1, 2, \dots, p\}$ ,  $\mathbf{x}^\alpha \in \mathbf{R}^n$ ,  $\mathbf{y}^\alpha \in \mathbf{R}^m$  a FS and  $\mathbf{M}$  its memory. Without loss of generality suppose that  $p$  is odd. Let  $\tilde{M}_i, i=1, m$  a row of matrix  $\tilde{\mathbf{M}}$  (and altered version of  $\mathbf{M}$ ) and  $\tilde{\mathbf{x}}^\alpha$  an altered version of a fundamental key pattern  $\mathbf{x}^\alpha$ . Let  $n$  be the number of elements of  $\tilde{M}_i$  and the number of elements of  $\tilde{\mathbf{x}}^\alpha$ . Let  $\tilde{M}_{ij}, j=1, n$  the  $j$ -th component of row  $\tilde{M}_i$ . If  $\tilde{M}_{ij} + \tilde{x}_j^\alpha \leq \frac{D-1}{2}$ , then  $y_i^\alpha = \tilde{M}_{ij} \diamond_B \tilde{x}_j^\alpha$ . If this holds for all  $i$  then  $\mathbf{y}^\alpha = \tilde{\mathbf{M}} \diamond_B \tilde{\mathbf{x}}^\alpha$ .

## 4 Experiments with Real Patterns

In this section, it is shown the applicability of the results given in section 3. Experiments were performed on different sets of images. In this paper we show the results obtained with photos of five famous mathematicians. These are shown in Fig. 1. The images are  $51 \times 51$  pixels and 256 gray-levels.

To build the memory, each image  $f_{51 \times 51}(i, j)$  was first converted to a pattern vector  $\mathbf{x}^\xi$  of dimension 2,601 ( $51 \times 51$ ) elements by means of the standard scan method, giving as a result the five patterns:

$$\mathbf{x}^\xi = [x_1^\xi \quad x_2^\xi \quad \dots \quad x_{2601}^\xi] \quad \xi = 1, \dots, 5.$$

It is not difficult to see that this set of vectors does not satisfy the conditions established by Theorem 1 and its Corollary. It is thus transformed into an auxiliary FS by means of the transformation procedure described in Sect. 2.3, giving as a result the transformed patterns:

$$\mathbf{z}^\xi = [z_1^\xi \quad z_2^\xi \quad \dots \quad z_{2601}^\xi] \quad \xi = 1, \dots, 5.$$

It is not difficult to see in the transformed domain, each transformed pattern vector is an additive translation of the preceding one.



**Fig. 1.** Images of the five famous people used in the experiments. (a) Descartes. (b) Einstein. (c) Euler. (d) Galileo, and (e) Newton. All Images are  $51 \times 51$  pixels and 256 gray levels.

First pattern vector  $\mathbf{z}^1$  was used to build matrix  $\mathbf{M}$ . Any other pattern could be used due to according to Corollary 1:  $\mathbf{M}^1 = \mathbf{M}^2 = \dots = \mathbf{M}^5 = \mathbf{M}$ . To build matrix  $\mathbf{M}$ , equations (1), (2) and (3) were used.

#### 4.1 Recalling of the Fundamental Set of Images

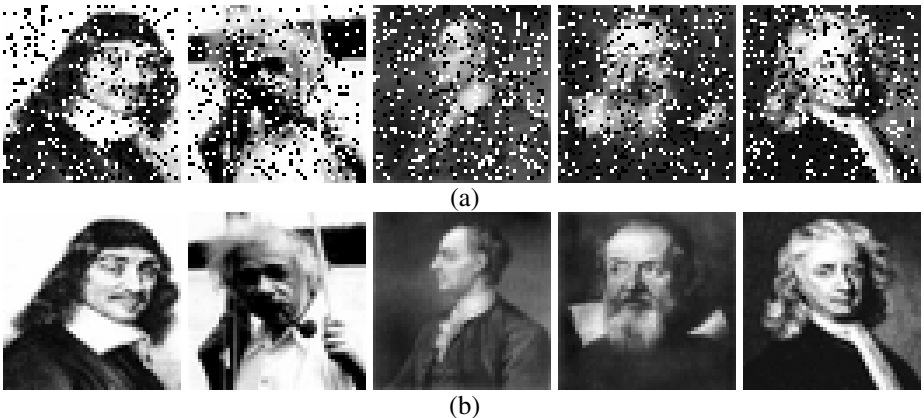
Patterns  $\mathbf{z}^1$  to  $\mathbf{z}^5$  were presented to matrix  $\mathbf{M}$  for recall. Equations (4) and (5) were used for this purpose. In all cases, as expected, the whole FS of images was perfectly recalled.

#### 4.2 Recalling of a Pattern from a Distorted Version of it

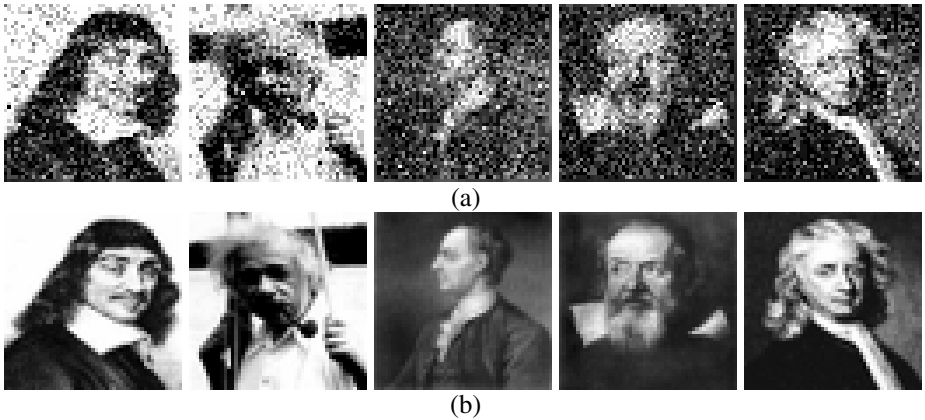
Two experiments were performed. In the first experiment the effectiveness of Proposition 2 was verified when less than 50% of the elements of the memory and of pixels of an image (taken together) were distorted by mixed noise. In the second experiment the effectiveness of Proposition 3 was verified when all the elements of the memory and all pixels of an image were distorted with noise but with absolute magnitude less than  $D/2$ . According to the material presented correct recall should occur in all cases.

##### 4.2.1 Effectiveness of Proposition 2

In this case, at random, less than 50% of the elements of the memory and of pixels of an image (taken together) were corrupted with saturated mixed noise. For each photo several noisy versions with different levels of salt and pepper noisy were generated. Figure 2(a) shows 5 of these noisy images. Note the level of added distortion. When applying the recalling procedure described in Sect. 2.3, as specified by Proposition 2 in all cases as shown in Fig. 2(b) the desired image was of course perfectly recalled.



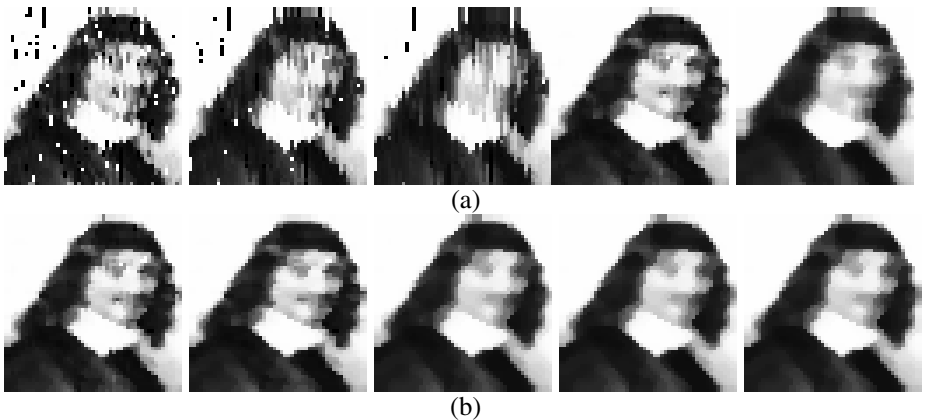
**Fig. 2.** (a) Noisy images used to verify the effectiveness of Proposition 2 when less than 50% of the elements of patterns and of the elements of the memory (taken together) are distorted by noise. (b) Recalled images.



**Fig. 3.** (a) Noisy images used to verify the effectiveness of Proposition 3 when the absolute magnitude of the noise added to the elements of the matrix and to values of the pixels of the patterns is less than  $D/2$ . (b) Recalled versions.

#### 4.2.2 Effectiveness of Proposition 3

In this case all elements of the five images shown in Fig. 1 were distorted with mixed noise but respecting the restriction that the absolute magnitude of the level of noise added to a pixel is inferior to  $D/2$ . For each image a noisy version was generated. The five noisy versions are shown in Fig. 3(a). When applying the recalling procedure described in Sect. 2.3, as expected in all cases the desired image was perfectly recalled. Figure 3(b) shows the recalled versions.



**Fig. 4.** (a) Filtered versions of Fig. 3(a) with a one-dimensional median filter of sizes  $1 \times 3$ ,  $1 \times 5$ ,  $1 \times 7$  and bi-dimensional median filter of  $3 \times 3$  and  $5 \times 5$ . (b) Filtered versions of Fig. 3(a) with a bi-dimensional median filter of size  $3 \times 3$  applied recursively five times.

#### 4.2.3 Results Obtained with a Linear Median Filter

In this section we present the results obtained when applying a one-dimensional and a bi-dimensional median filter of different sizes to one of the images of Fig. 2(a), in this

case to the first image. Figure 4(a) shows the filtered images with a one-dimensional median filter of sizes 3, 5 and 7, and with a bi-dimensional median filter of sizes  $3 \times 3$  and  $5 \times 5$ . Figure 4(b) shows the results obtained with a bi-dimensional median filter of size  $3 \times 3$ , applied recursively 5 times to the image. As can be appreciated in neither of the cases the noise introduced to the image has been completely eliminated. Note also how in the case of the median filter of size  $3 \times 3$ , applied recursively the result changes very little as the filter is applied. It can also be appreciated that in general better results are obtained with a bi-dimensional median filter than with a one-dimensional median filter.

## 5 Conclusions

In this paper we have studied how the performance of a median associative memory is affected when the values of its elements are altered by mixed noise. We have provided with formal conditions under which the memory is still able to correctly recall a pattern of the fundamental set of patterns either from a non-altered or a noisy version of it. Experiments with real patterns were also given to show the efficiency of the proposal. From the experiments it is also shown that the proposal performs better than the ordinary 1-D and 2-D mean filtering masks, either applied one time or iteratively to the image.

**Acknowledgements.** The authors give thanks to CIC-IPN and COFAA-IPN for the economical support. Authors also thank SIP-IPN and CONACYT for their support under grants 20060517, 20071438 and 46805. We thank also the reviewers for their comments for the improvement of this paper.

## References

1. Steinbuch, K.: Die Lernmatrix, *Kybernetik*, (1961) 1(1):26-45.
2. Anderson, J. A.: A simple neural network generating an interactive memory, *Mathematical Biosciences*, (1972) 14:197-220.
3. Hopfield, J. J.: Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences*, (1982) 79: 2554-2558.
4. Ritter, G. X. *et al.*: Morphological associative memories, *IEEE Transactions on Neural Networks*, (1998) 9:281-293.
5. Sossa, H., Barrón, R. and Vázquez, R. A.: New Associative Memories to Recall Real-Valued Patterns. LNCS 3287. Springer Verlag (2004) 195-202.
6. Sossa, H., Barrón, R. and Vázquez, R. A.: Transforming Fundamental Set of Patterns to a Canonical Form to Improve Pattern Recall. LNAI 3315. Springer Verlag. (2004) 687-696.
7. Sossa, H. and Barrón, R.: Median Associative Memories: New Results. LNCS 3773. Springer Verlag. (2005) 1036-1046.
8. Sossa, H., Barrón, R., Cuevas, F. and Aguilar, C.: Associative gray-level pattern processing using binary decomposition and  $\alpha\beta$  memories. *Neural Processing Letters* (2005) 22:85-11.
9. B. Cruz, H. Sossa, and R. Barrón (2007): Associative processing and pattern decomposition for pattern reconstruction. *Neural Processing Letters* 25(1):1-16.
10. Sossa, H. and Barrón, R.: Extended  $\alpha\beta$  associative memories. To appear in *Revista Mexicana de Física* (2007).

# Impact of Learning on the Structural Properties of Neural Networks

Branko Šter, Ivan Gabrijel, and Andrej Dobnikar

Faculty of Computer and Information Science, University of Ljubljana  
Tržaška 25, 1000 Ljubljana, Slovenia  
`branko.ster@fri.uni-lj.si`

**Abstract.** We research the impact of the learning process of neural networks (NN) on the structural properties of the derived graphs. A type of recurrent neural network is used (GARNN). A graph is derived from a NN by defining a connection between any pair of nodes having weights in both directions above a certain threshold. We measured structural properties of graphs such as characteristic path lengths ( $L$ ), clustering coefficients ( $C$ ) and degree distributions ( $P$ ). We found that well trained networks differ from badly trained ones in both  $L$  and  $C$ .

## 1 Introduction

After the first theoretical studies of random graphs by Erdos and Renyi [1], complex networks from the real world became a target of numerous investigations [2,3,4,5,6]. The main question was whether systems such as the World Wide Web, the Internet, chemical networks and neural networks follow the rules of random networks or related graphs. The structural properties of graphs are usually quantified by characteristic path lengths, clustering coefficients and degree distributions [3,5]. Various types of graphs are classified based on the values of these parameters [6]. Besides regular and random graphs with two extreme topologies, two new topologies and consequently two new types of graphs in between were identified: Small-World (SW) topology [2], according to the small average length between the nodes (small-world feature) and Scale-Free (SF) topology [6] due to the exponential degree distribution. It was further recognized that the evolution of real networks and their topologies are governed by certain robust organizing principles [3]. This implies that there is a topological difference between organized and non-organized (or random) networks. The aim of this paper is to show the influence of learning on the structural parameters of neural network-related graphs.

After a short presentation of the background theory of complex systems and neural networks, the experimental work is described together with results showing how the learning process of neural networks changes the structural properties of the related graphs.



## 2 Background Theory

### 2.1 Complex Systems

Complex systems describe a wide range of systems in nature and society [6]. Structurally, they can usually be viewed as networks.

Graphs are usually used to represent a complex connection of units (network), described by  $G = (N, E)$ , where  $N$  is a set of  $n$  nodes (units or cells) and  $E$  is a set of  $e$  edges, and where each edge connects two units. There are topological differences between the graphs that correspond to the regular, random or completely random connections. To randomize connections, we start from a regular lattice with  $n$  nodes and  $k$  edges per vertex and rewire each edge at random with probability  $p$ .

The structural properties of graphs are quantified by characteristic path length  $L(p)$ , clustering coefficient  $C(p)$  and degree distribution  $P(k)$  [3].

$L(p)$  measures the typical separation between two nodes on the graph, where the lengths of all edges are 1. Such graphs are relational, and the measure describes a global property.

$C(p)$  shows the cliquishness of a typical neighbourhood, or average fraction of existing connections between the nearest neighbours of a vertex, which is a local property.

$P(k)$  is the probability that a randomly selected node has exactly  $k$  edges.

Besides regular networks or graphs (lattices with  $n$  vertices and  $k$  edges per vertex), there are three main groups of networks (graphs) that differ in their structural properties or topologies:

- Random networks
- Small-World networks
- Scale-Free networks

**Random Networks (RNs).** A random network is obtained from a regular one by rewiring each edge at random with probability  $p$ . This construction allows us to 'tune' the graph between regularity ( $p = 0$ ) and disorder ( $p = 1$ ).

Random networks have a fixed number of nodes or vertices ( $n$ ). Two arbitrary nodes are connected with probability  $p$ . On average, the network therefore contains  $e = pn(n - 1)/2$  edges. The degree distribution  $P(k)$  is a binomial, so the average degree is:  $\langle k \rangle = p(n - 1) \sim pn$  for large  $n$ .

The estimate for an average shortestpath length is obtained from:

$$\langle k \rangle^L = n, \quad (1)$$

$$L(p) = \frac{\ln(n)}{\ln(\langle k \rangle)} \sim \frac{\ln(n)}{\ln(pn)}, \quad (2)$$

which is low, typical for the small-world effect. The clustering coefficient is:

$$C(p) = p \sim \frac{\langle k \rangle}{n}, \quad (3)$$

since the edges are distributed randomly. It is much smaller than in a comparable real world network with the same number of nodes and edges.

**Small-World Networks (SWNs).** SWNs can be constructed from ordered lattices by random rewiring of edges ( $p$  is typically around 0.1-0.3) or by the addition of connections between random vertices.

SWNs have small average shortest-path lengths, like RNs, but much greater clustering coefficients than RNs. SWNs fall between regular and random graphs and can be treated analytically. The shape of the degree distribution is similar to that of random graphs. It has a peak at  $\langle k \rangle$  and decays exponentially for large and small  $k$ . The topology is relatively homogenous, all nodes having approximately the same number of edges.

**Scale-Free Networks (SFNs).** The degree distributions of SFNs have power-law tails:

$$P(k) \sim k^{-\gamma}. \quad (4)$$

This sort of distribution occurs if the random network grows with preferential attachment. This means that a node with  $m$  edges is added at every time step ( $m \leq m_0$ , where  $m_0$  is the starting number of nodes) and connected to an existing node  $i$  with probability:  $p(k_i) = k_i / \sum k_j$ , which implies that it is more probably connected to a node with more edges than to one with fewer edges. It has been shown that many large networks in the real world (WWW, Internet, scientific literature, metabolic networks, etc.) exhibit a power-law degree distribution and therefore belong to the SF type of networks.

## 2.2 Generalized Architecture of Recurrent Neural Networks (GARNN)

There are many types of artificial neural networks with different topologies of connections between neurons, mathematical models of neurons and learning algorithms [9]. There are preferential tasks for each type of neural network, each with advantages over the other types. For example, if one wants to use an artificial neural network for the classification/identification of an unknown dynamic system, then a network with feedback connections is needed. For the purpose of this paper, only fully connected neural networks (each neuron connected to every other neuron) are relevant. One of them is Hopfield neural network, with a rather simple learning algorithm (following the rule that the coupling strength equals the sum of the corresponding component values of the patterns to be stored). However, it is not efficient for the purpose of classification/identification of a general dynamic system. A classical recurrent neural network, on the other hand, uses a gradient-based learning algorithm RTRL (Real Time Recurrent Learning), which is not only very time consuming but often does not converge fast enough if the network is large and the task relatively difficult.

For the purposes of this paper, a GARNN is chosen [7], for the following reasons: a) it has a general topology (with two layers of neurons, where each neuron is connected to every other in the network), b) it uses a special learning

algorithm that demonstrates good convergence characteristics (mostly due to the modified RTRL, the neuron level of Decoupled Extended Kalman Filter for updating weights and alpha-projected teacher forcing technique). Unfortunately, the combination of the GARNN and the learning algorithm doesn't allow one to use a large number of neurons expected of a complex system. For the purposes of the experimental work, we used much larger neural networks than are actually needed for the selected application. This can be biologically justified by the way that living organisms also use a huge number of neurons for simple functions performed at a particular moment. What we want to show is that through the learning process of the GARNN responsible for the organizing evolution of network weights, the structural parameters of the related graphs change from the features of a RN to those of SWNs with a different degree distribution. We see no reasonable argument that this should not be the same in complex neural networks with a large number of neurons.

The block diagram of the GARNN is shown in Fig. 1. There are three layers in the network: an input layer, a hidden layer and an output layer. The number of neurons in the output layer is equal to  $m + r$  (known as  $m + r$  heuristics [7]), where  $m$  is the number of outputs from the network and  $r$  is the number of neurons in the hidden layer.

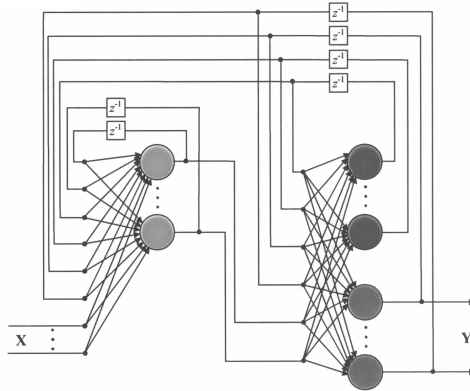


Fig. 1. Generalized architecture of recurrent neural networks (GARNN)

### 3 Experimental Work

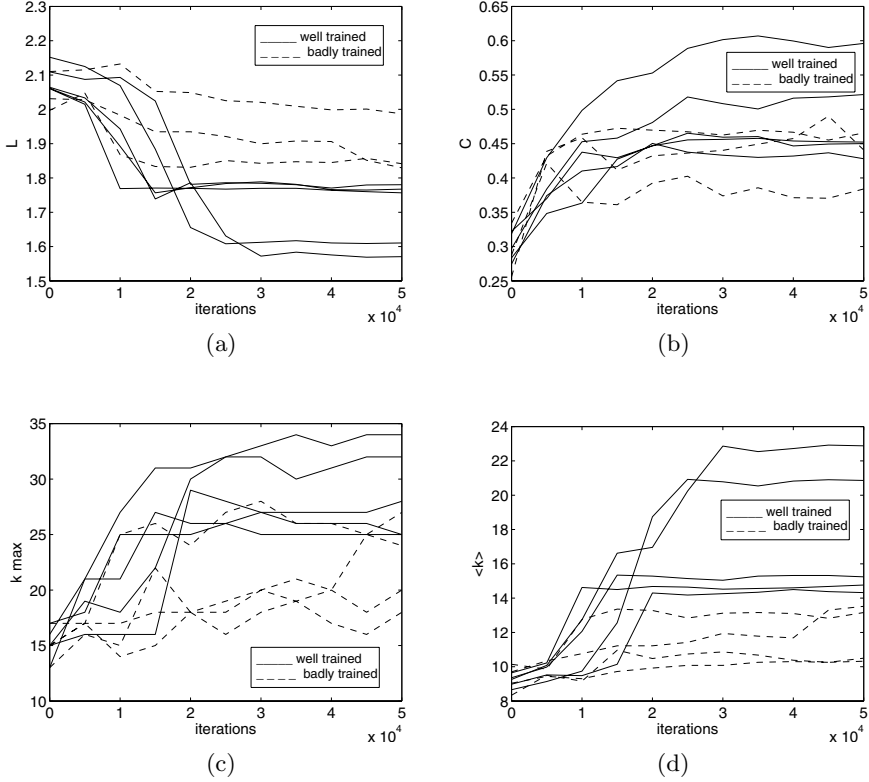
We want to show experimentally the influence of the learning algorithm of the GARNN, as one of the possible organizing evolutions, on the topological parameters of the related graphs. The GARNN was trained to identify an unknown discrete dynamic system, which in our case is a finite state machine which performs the time-delayed XOR( $d$ ) function of two subsequent inputs, delayed by  $d = 4$ . It has a state transition diagram with 64 states, which makes the identification task to be performed by the GARNN non-trivial. Though the problem

can be solved successfully with a GARNN with only 34 neurons [7], we increased the number to 50 and 80. We have done also an experiment with 100 neurons on the task XOR(0). Even this number might be disputable from the view-point of complex systems. But even so, the learning process should find the weights that are compatible within the application. On the other hand, the size of the GARNN is not very much smaller than the number of neurons in *C.elegans* (282), a swarm which has recently been investigated many times [28], or the number of neurons in previous experiments with Hopfield networks or Cellular Neural Networks (280, 96) with trivial learning algorithms [10,11]. For efficient learning, the supervised learning algorithm of the GARNN needs a training sequence of over 150,000 random input binary characters and additional 15,000 for the testing purposes. This learning process takes more than a week on a cluster (16 PCs at 3.5GHz), and the time grows exponentially with the increasing number of neurons. This is the main reason we had to limit the sizes of GARNNs used in our experimental work.

We performed the experiment as follows: a GARNN of sizes 50, 80 and 100 is trained on the identification task of XOR ( $d = 4$  and  $d = 0$ ). After every 5,000 (8,000 and 10,000 at 80 and 100 neurons, respectively) learning steps the weights are saved for the purpose of structural analysis. The experiment is repeated 10 times with different initial weights (chosen randomly between -0.5 and +0.5). After every 5,000 (8,000, 10,000) learning steps, all three structural parameters  $L$ ,  $C$  and  $P$  are calculated. Before this calculation, however, a transformation from the GARNN to a unidirectional graph is made. It is performed with the help of a threshold parameter as follows. Each weight in the GARNN is compared with the threshold, and if the absolute values of both weights from node  $i$  to node  $j$  and vice versa are greater than the threshold, these two nodes are connected with the distance 1 (relational type of graph), otherwise they are disconnected (distance is 0). Self- and multiple links are disallowed, so  $w_{ii} = 0$  for all  $i$ . The threshold value is chosen with the help of additional tests on the trained GARNN. If all the weights with absolute values lower than the threshold are neglected in the network (replaced with 0), the performance of the GARNN on the task should not decrease.

First, 10 experiments with a GARNN of 50 neurons were performed. Two experiments were left out: one network was only moderately trained (others were either well trained or badly trained), while another (badly trained) had isolated groups of neurons, which made the calculation of  $L$  impossible. Figure 2a shows the average shortest-path lengths  $L$  versus training iterations on the delayed XOR ( $d = 4$ ). The successful training experiments are shown with solid lines, while the unsuccessful training experiments (hit ratio less than 100%) are shown with dashed lines. It is obvious that successfully trained networks have lower values of  $L$  (after 50,000 iterations) than unsuccessfully trained networks.

Figure 2b shows the course of the clustering coefficient  $C$  in the same experiments.  $C$  is obviously increasing during training. Besides, the well-trained networks have higher clustering than the unsuccessfully trained. However, the difference is not as pronounced as with  $L$ .



**Fig. 2.** (a) Average shortest-path length  $L$ , (b) clustering coefficient  $C$ , (c) maximal and (d) mean values of  $k$  versus training iterations on the delayed XOR ( $d = 4$ ) with a 50-neuron GARNN

Instead of plotting degree distributions  $P(k)$  at different times during training, we merely present the course of maximal and average values of degree  $k$  (Fig. 2c and Fig. 2d). One network was left out (moderately trained). As expected, degrees are increased during the course of training. However, the degrees, especially the average  $\langle k \rangle$ , are clearly larger in the well trained networks than in the badly trained ones.

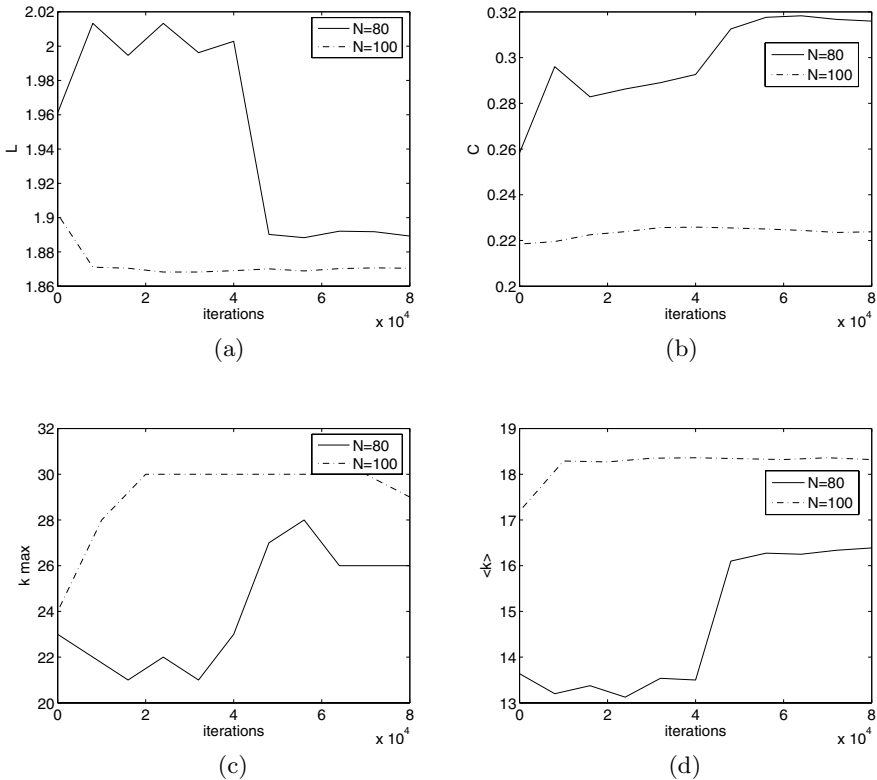
The values of  $L$ ,  $C$ ,  $k_{max}$  and  $\langle k \rangle$  averaged over the well trained and over the badly trained networks are shown in Table I.

Next, we made two experiments with larger numbers of neurons: first with 80 neurons on the same task and the second with 100 neurons on task XOR ( $d = 0$ ). There was only one run in each, due to the very long training time. Both training experiments were successful. Figure 3a and Fig. 3b show  $L$  and  $C$  during training. The trends are similar to those of the previous experiments.

The course of maximal and average values of degree  $k$  are shown in Fig. 3c and Fig. 3d.

**Table 1.** Average values of  $L$ ,  $C$ ,  $k_{max}$  and  $\langle k \rangle$  over well trained and over badly trained networks

iter.	0	5000	10000	15000	20000	25000	30000	35000	40000	45000	50000
L (good)	2.09	2.06	1.95	1.83	1.75	1.72	1.71	1.71	1.70	1.70	1.70
L (bad)	2.05	2.06	1.99	1.94	1.94	1.93	1.92	1.92	1.92	1.90	1.89
C (good)	0.30	0.38	0.43	0.45	0.48	0.49	0.49	0.49	0.49	0.49	0.49
C (bad)	0.29	0.43	0.43	0.41	0.43	0.44	0.43	0.43	0.43	0.43	0.43
$k_{max}$ (good)	15.2	19.0	21.4	24.2	28.2	28.8	28.8	28.4	28.4	28.8	28.8
$k_{max}$ (bad)	15.0	16.7	17.7	20.2	19.5	20.0	21.5	21.2	20.7	21.0	22.2
$\langle k \rangle$ (good)	9.2	9.8	11.7	13.8	16.0	17.0	17.5	17.4	17.6	17.6	17.6
$\langle k \rangle$ (bad)	9.3	9.8	10.5	11.3	11.2	11.3	11.5	11.5	11.4	11.7	11.9

**Fig. 3.** (a) Average shortest-path length  $L$  and (b) clustering coefficient  $C$  versus training iterations on XOR (4) with an 80-neuron GARNN and on XOR (0) with a 100-neuron GARNN. (c) Maximal and (d) mean values of  $k$  versus training iterations on XOR(4) with an 80-neuron GARNN and on XOR (0) with a 100-neuron GARNN.

## 4 Conclusion

The paper deals with generalized recurrent neural networks (GARNN) as a special type of complex system from the real world. The main goal was to show the influence of the learning process on the structural parameters of the related graphs. With the help of experiments from the area of the identification of dynamic systems ( $XOR(d)$ ) we show that the learning process changes the parameters of the graphs of the GARNNs from the RNs towards the SWNs. The global parameters  $L$  are decreasing and the local parameters  $C$  increasing, which is typical for SWNs. The degree distributions  $P(k)$  are moving towards modified distributions with increased highest degrees and higher average degrees. This means that the learning of the GARNNs changes the topologies of the related graphs in the direction of SWNs. The learning of neural networks is therefore one of many robust organizing principles, observed in nature, that changes complex structures into a special type of small-world networks.

## References

1. Erdos, P., Renyi, A.: On random graphs. *Publicationes Mathematicae* **6** (1995) 290–297
2. Watts, D. J.: *Small Worlds*. Princeton university press (1999)
3. Reka, A., Barabasi, A. L.: Statistical mechanics of complex networks. *Reviews of modern physics* **74** (2002) 47–97
4. Watts D. J., Strogatz, S. H.: Collective dynamics of 'small-world' networks. *Letters to nature* **393/4** (1998) 440–442
5. Dorogovtsev S. N., Mendes, J. F. F.: Evolution of networks. *Advances in physics* **51-4** (2002) 1079–1187
6. Bornholdt S., Schuster, H. G.: *Handbook of Graphs and Networks*. Wiley-VCH (2003)
7. Gabrijel, I., Dobnikar, A.: On-line identification and reconstruction of finite automata with generalized recurrent neural networks. *Neural Networks* **16** (2003) 101–120
8. Kim, J. B.: Performance of networks of artificial neurons: The role of clustering. *Physical Review E* **69** (2004) 045101/1-4
9. Haykin S.: *Neural Networks*. Prentice-Hall (1999)
10. McGraw, P.N., Menzinger, M.: Topology and computational performance of attractor neural networks. *Physical Review E* **68** (2003) 047102/1-4
11. Torres, J. J., Munoz, M. A., Marro, J., Garrido, P. L.: Influence of topology on the performance of a neural network. *Neurocomputing* **58-60** (2004) 229–234

# Learning Using a Self-building Associative Frequent Network

Jin-Guk Jung<sup>1</sup>, Mohammed Nazim Uddin<sup>1</sup>, and Geun-Sik Jo<sup>2</sup>

<sup>1</sup> Intelligent e-Commerce Systems Laboratory,  
253 Yonghyun-Dong, Nam-Gu, Incheon, Korea 402-751  
{gj4024, nazim}@ieslab.inha.ac.kr

<sup>2</sup> School of Computer Engineering, Inha University,  
253 Yonghyun-Dong, Nam-Gu, Incheon, Korea 402-751  
gsjo@inha.ac.kr

**Abstract.** In this paper, we propose a novel framework, called a *frequent network*, to discover frequent itemsets and potentially frequent patterns by logical inference. We also introduce some new terms and concepts to define the *frequent network*, and we show the procedure of constructing the *frequent network*. We then describe a new method *LAFN* (Learning based on Associative Frequent Network) for mining frequent itemsets and potentially patterns, which are considered as a useful pattern logically over the *frequent network*. Finally, we present a useful application, classification with these discovered patterns from the proposed framework, and report the results of the experiment to evaluate our classifier on some data sets.

## 1 Introduction

Since the Hebbian Learning Rule was first proposed by Canadian psychologist Donald Hebb [4], understanding the mechanism to recall facts and information by invoking a chain of associations within a complex web of embedded knowledge or stored patterns in the human brain remains a challenge in the area of Neural Networks (NN). Many researchers have proposed efficient methods for resolving this problem using Associative Neural Networks (ANN). Amari [2] and Hopfield [5] presented interesting results demonstrating that this network has associative properties.

The ANN provides storage of information within which each neuron stores parts of information needed to retrieve any stored data record. It also has the property of parallel operation as the artificial neural networks. Moreover, the network can be trained using accumulated experience. However, the structure of the network and the mechanism of learning are somewhat complicated.

In this paper, we introduce the Associative Frequent Network (AFN), a weighted network to store patterns. The network consists of simple interconnected neurons/vertices and synapses/edges for storing projected information from a given training set. Neurons store only the count of occurrences of an event. The relationship between events is represented as a weighted edge.



We show that the proposed AFN, while simple and intuitive in structure, is useful. The method to construct the associative frequent network is explained and we also describe how the network can be learned from previous experience.

The remainder of this paper is organized as follows. In Sect. 2, we introduce a new framework, the *frequent network*, with new concepts and an algorithm to construct the network for an associative neural network. Section 3 explains how to find frequent patterns included frequent itemsets and logical patterns. Section 4 describes how to learn associations with the *frequent network*. We also present an exploration method for the *frequent network* constructed in the previous section to solve the problem described in Sect. 2. In Sect. 5, we present a case study in which we tested our methods on data sets. Conclusions are presented in the final section.

## 2 The Frequent Network: Design and Construction

We consider a weighted network of neurons/vertices and *synaptic* junctions/edges operating in parallel. The formal neurons in our work are elements, which have a deterministic threshold, with several binary inputs and outputs. More precisely, the formal neuron  $i$  has  $k$  binary inputs  $s_j$ ,  $j = 1, 2, \dots, k$ , which can be the outputs of the several neurons of the network. If any such neuron  $i$  is *active*, its output value, which is denoted by the variable  $o_j$ , is fired to connected neurons. Otherwise, the neuron is *inactive*. That is, its output cannot have an influence on connected neurons. The *synaptic* junction of neuron  $i$  receiving information from an adjacent neuron  $j$  is represented by a coupling coefficient,  $e_{ij}$ . The value of the junction can be either 1, if the junction is *active* or 0 if the junction is *inactive*.

### 2.1 The Associative Frequent Network

We assume that the associative frequent network with vertices and edges shape to be a weighted network. Vertices/neurons in this network store only the count of occurrences of an event. The relationship between events is represented by a weighted edge that also stores the count of simultaneous occurrences of two events. We first define some concepts for our study.

**Definition 1.** *The **Associative Frequent Network** is defined in the following way:*

1. *It consists of a set of vertices and a set of edges.*
2. *A vertex exists in the network corresponding to an event. The vertex has three attributes, **name**, **count**, and **edges**, where **name** represents the name of the event, **count** stores accumulated occurrences of the event until the training is finished, and **edges** is the set of edges connected with this vertex. Each vertex has at least two edges. The first edge is for input of the signal and the other is for output of the neuron.*
3. *An edge exists in the network corresponding to a synaptic junction. The edge has three attributes, **fromVertex**, **toVertex**, and **count**, where **fromVertex** and **toVertex** link to vertices and **fromVertex** occurs earlier than **toVertex**, and **count** stores accumulated occurrences of events together.*

A path  $(a, c)$  in the network is a sequence of vertices such that from each of its vertices there is an edge to the next vertex in the sequence. The first vertex is called the *start vertex* and the last vertex is called the *end vertex*. The other vertices in the path are called *internal vertices*.

**Definition 2 (path-connected).** Let  $V$  be a set of vertices in a frequent network, and  $X = \{x \mid x \in V, x \in (a, c)\}$  be a set of vertices in the path  $(a, c)$ . Then,  $X$  is said to be **path-connected** if for any two vertices  $x$  and  $y$  in  $X$  there exists the edge  $xy$ .

For example, if we have a frequent network that has 3 vertices,  $a$ ,  $b$ , and  $c$ , and 2 edges,  $ab$  and  $bc$  then the path  $(a, c)$  is not **path-connected** because the edge  $ac$  does not exist.

**Definition 3 (countfold).** Let  $X$  be a set of vertices in a frequent network and  $x$  be a vertex in  $X$ . Let  $y$  also be a vertex in  $X$ ,  $y \neq x$ , and  $xy$  be the edge starting from  $x$  to  $y$ . Both  $x.count$  and  $xy.count$  are the count of the vertex  $x$  and the edge  $xy$ , respectively. A **countfold** of any vertex  $x$  in the path  $X$ , which is denoted by  $countfold(x)$ , is the number that indicates how many times edges  $E$  which are connected to  $x$  are folded together. It is calculated by the equation below.

$$countfold(x) = \sum_{xy \in E} xy.count - x.count \quad (1)$$

For example, suppose there exists a frequent network that has 3 vertices,  $a$ ,  $b$ , and  $c$ , and 3 edges,  $ab$ ,  $ac$ , and  $bc$ . The counts of each vertex are  $a.count = 1$ ,  $b.count = 1$ , and  $c.count = 1$ , and the count of each edge are  $ab.count = 1$ ,  $ac.count = 1$ , and  $bc.count = 1$ . Then, the **countfold** of the vertex  $b$  is 1 by (1). However, if the count of vertex  $b$  is 2, the **countfold** of the vertex  $b$  becomes 0.

**Definition 4 (path support).** Let  $f(a, c)$  be the **path support** from vertex  $a$  to vertex  $c$  in a frequent network. The **path support** is an integer function  $f : V \times V \rightarrow \mathbb{N}$ .

$$f(a, c) = \max\{\min\{xy.count \mid xy \in E\}, \max\{countfold(x) \mid x \in V\}\}, \quad (2)$$

where for any vertices  $x$  and  $y$ ,  $x \neq y$ , in the path  $(a, c)$ ,  $V$  and  $E$  are a set of vertices and edges of the path in the network respectively.

## 2.2 Frequent Network Construction and Example

In this section, we present an example and an algorithm to construct a frequent network based on Definition 1. The algorithm needs only one scan of the transaction database  $\mathbf{D}$ . During this scan the minimum support threshold is not needed. The complexity of this algorithm depends on the number of transactions in  $\mathbf{D}$ , the number of items, and the density of relationships among each item. The algorithm is presented in the following.

**Algorithm. F-network construction****Input:** Database,  $\mathbf{D}$ , of transactions;**Output:**  $\mathbf{N}$ , frequent network projected  $\mathbf{D}$ .**Method:** Call *buildingFN*( $\mathbf{D}$ ).**Procedure** *buildingFN*( $\mathbf{D}$ )

```

1:  {
2:    initialize a frequent network,  $\mathbf{N}$ ;
3:    for each transaction  $T \in \mathbf{D}$  {
4:       $oT = \text{sort}(T)$ ; // ordering items lexicographically.
5:       $\mathbf{N}.\text{insertTrans}(oT)$ ; // insert the transaction  $oT$  into F-network.
6:    }
7:    return  $\mathbf{N}$ ;
8:  }
```

**Procedure** *insertTrans*( $oT$ )

```

1:  {
2:    for each vertex  $x \in oT$  {
3:      if( $x \in \mathbf{N}$ ) increment  $x.\text{count}$ ;
4:      else {
5:        create a new vertex  $v$  corresponding to  $x$ ;
6:        set its itemName related to  $x$  and count to 1;
7:        put  $v$  into vertices;
8:      }
9:    }
10:   for each edge  $xy \subset oT$  {
11:     if( $xy \subset \mathbf{N}$ ) increment  $xy.\text{count}$ ;
12:     else {
13:       create a new edge  $e$  corresponding to  $xy$ ;
14:       set  $e.\text{count} = 1$ , fromVertex and toVertex link to
15:       the start vertex and end vertex respectively;
16:       put  $e$  in the edges of the start vertex
17:       put  $e$  into edges;
18:     }
19:   }
```

*Example 1.* Let the transaction database  $\mathbf{D}$  be the left part of Fig. [1](#). We can construct a frequent network projected transaction database  $\mathbf{D}$ . After reading each transaction, the algorithm inserts some itemsets into the network and stores the count of each item and each 2-itemsets. the constructed frequent network is shown in the right side of Fig. [1](#).

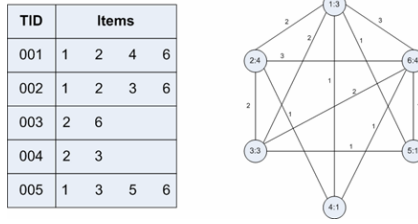


Fig. 1. The network transformed database

### 3 Mining Frequent Itemsets Using *F-network*

In the proposed network structure, we observe a limitation to mine frequent patterns. We define a new term to solve this limitation and extend the meaning of frequent itemsets used in the data mining field.

By using *countfold*, we are able to compute the support value of a  $k$ -itemset ( $k > 2$ ) such as  $\{u, v, w\}$ . However, as shown in Fig. 2, we cannot compute the exact *support* of a 3-itemset  $\{1, 2, 3\}$  by (1). In other words, there are no transactions including the 3-itemset  $\{1, 2, 3\}$  in the database, while in the transformed network the path support of the itemset is 2 by (2). To consider this case, we define a logical pattern and extend the meaning of the traditional frequent itemset.

**Definition 5 (Logical Pattern).** Let  $(a, c)$  be a path. We call the path a *logical pattern* if the path is path-connected and the *pathSupport* of the path is greater than or equal to the given minimum support threshold,  $\Theta$ . However, the path is not actually a frequent itemset.

For example, suppose that we have six transactions, as shown in Fig. 2, and we can transform this database into a frequent network, as shown in the right side of Fig. 2. In this case, the itemset  $\{1, 2, 3\}$  is not actually a frequent itemset and has a support threshold,  $\Theta = 2$ . However, this is considered to be a logically frequent pattern, since the pattern is path-connected and satisfies the threshold.

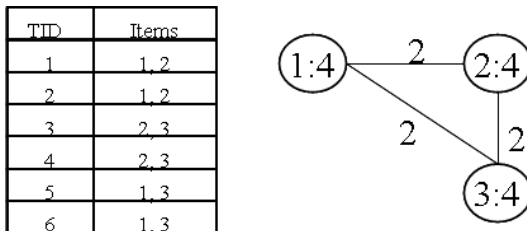


Fig. 2. The logical pattern

**Definition 6 (Frequent Itemset).** *In a frequent network, an itemset is **frequent** if the **path support** of the path corresponding to the itemset is greater than or equal to the minimum support threshold,  $\Theta$ .*

When we find frequent itemsets in the given  $F$ -network, we have to check the path in the following way.

1. Select the minimum count of an edge after checking the counts of all edges in the path.
2. Return the new matrix to calculate countfold in the next step.
3. Return the real path support.

The following describes an algorithm to find frequent patterns from the given network. The algorithm searches frequent patterns with a depth first strategy.

**Algorithm. Mining frequent itemsets with F-network**

**Input:** F-network,  $\mathbf{N}$ , constructed by the previous algorithm and a minimum support threshold  $\Theta$ .

**Output:** The complete set of frequent itemsets.

**Method:** Call *miningFN*( $F$  - network,  $\Theta$ ).

**Procedure** *miningFN*( $F$  - network,  $\Theta$ )

```

1:  {
2:    remove some vertices and edges which do not satisfy  $\Theta$ .;
3:    for each vertex  $v_i \in \mathbf{N}$  {
4:      initialize  $path(v_i)$ ;
5:       $L_i = findLargeItemsets(F - network, path, \Theta)$ ;
6:    }
7:    return  $\cup_i L_i$ ;
8:  }
```

**Procedure** *findLargeItemsets*( $F$  - network,  $path$ ,  $\Theta$ )

```

1:  {
2:     $E = \{xy \mid edges \text{ in the last vertex } \}$ ;
3:    if  $E = \emptyset$  do
4:       $\{ L_i \cup path \mid pathSupport(path) \geq \Theta \}$ ;
5:    else for each edge  $xy \in E$  do {
6:      if  $xy.count \geq \Theta$  do {;
7:        initialize a new path included the previous path;
8:         $findLargeItemsets(F - network, newPath, \Theta)$ ;
9:      }
10:    }
11:     $\{ L_i \cup path \mid pathSupport(path) \geq \Theta \}$ ;
12:  }
```

## 4 Associative Classification Based on Frequent Network

Suppose a data item  $I = (i_1, \dots, i_n)$  with attributes  $A_1, \dots, A_n$ . Attributes can be categorical or continuous. For categorical attributes it is assumed that all the

possible values are mapped to set of consecutive positive integers. For a continuous attribute its value range is divided into some intervals, and the intervals are also mapped to consecutive positive integers.

Let  $C = (c_1, \dots, c_m)$  be a finite set of class labels to classify. A training data set is considered as a set of data items such that there exists a class label  $c_i \in C$  associated with it. For a given data set  $D$ , the *F-network* is constructed according to the procedure outlined in Sect. 2, where vertices hold the items  $I$  and also a class label  $c_i \in C$ . After constructing the network, a frequent itemset is found from the *F-network* according to the algorithm described in Sect. 3. Basically, that frequent itemset consists of associative items and a class label.

## 5 Experimental Evaluation

In this section, we present experiments to evaluate the efficiency of the proposed framework. We used 10 datasets from UCI ML Repository [9]. In our experiments, the minimum support is set to 1%, since accuracy heavily depends on the minimum support threshold value. According to previous studies on associative classification, *minsup* has a strong effect on the quality of the produced classifier [7,8]. Therefore, if *minsup* is set too high, classifiers may fail to cover all the training cases. From these observations, we follow the direction of previous works and set *minsup* to 1%. Note that the minimum confidence does not have a relatively strong effect on the quality of the classifier. Therefore, following a previous work [8], we set the minimum confidence value at 50%.

For the experiment, we first discretized attributes, the range of which has continuous values in the datasets, into intervals using the Entropy method in [3], and the intervals were also mapped to consecutive positive integers. For this purpose, we used the code taken from the MLC++ machine learning library [6]. All the error rates on each dataset were obtained from 10-fold cross-validations. The experimental results are shown in Table 1.

For comparison of the present experimental results, we considered three other popular classification methods, C4.5 [10], CBA [8], and CMAR [7], along with their default values. We compared the performance of our algorithm (*LAFN*) in terms of accuracy. As shown in Table 1, the average accuracy of *LAFN* is significantly higher than that of the three other methods, C4.5, CBA, and CMAR, respectively. Additionally, out of 10 data sets, *LAFN* maintains the best performance in 5 cases, and in 3 cases *LAFN* achieved average accuracy. Overall, the results indicate that *LAFN* outperforms the existing methods in accuracy.

In general, our method is based on a *frequent network* consisting of simple interconnected vertices and edges for storing projected information from given data sets. In addition, our classifier exploits a set of extended frequent itemsets, which can be said to be a superset of traditional itemsets because they are combined traditional frequent itemsets and logical patterns, discovered from the proposed *frequent network*. Therefore, we can obtain better performance with respect to accuracy, as shown in Table 1.

**Table 1. Comparison of C4.5,CBA,CMAR and *LAFN* on Accuracy**

Data set	Attr.	Cls.	Rec.	C4.5	CBA	CMAR	<i>LAFN</i>
AUTO	25	7	205	80.1	78.3	78.1	80.3
CLEVE	13	2	303	78.2	82.8	82.2	83
Diaetes	8	2	768	74.2	74.5	75.8	74.5
Glass	20	2	1000	72.3	73.4	74.9	73.2
LED7	7	10	3200	73.5	71.9	72.5	71.8
LYMPH	18	4	148	73.5	77.8	83.1	83.8
PIMA	8	2	768	75.5	72.9	75.1	75.6
SONAR	60	2	208	70.2	77.5	79.4	79.6
VEHICLE	18	4	846	72.6	68.7	68.8	70.5
WAVEFORM	21	3	5000	78.1	80	83.2	78

## 6 Conclusions

In this paper, we have presented a new algorithm (*LAFN*) for efficient enumeration of frequent itemsets based on a new framework, called a *frequent network*. The frequent network offers several advantages over other approaches: It constructs a highly compact frequent network, which is usually substantially smaller than the original database, and thus requires fewer costly database scans in the subsequent mining processes.

A key contribution of this paper is the provision of a new data structure that is efficient in storing information by only one scan. In addition, we expanded the meaning of the traditional frequent itemset to include logical patterns and described a novel method to compute *support* of  $k$ -itemsets directly for discovering frequent patterns from the *F-network*. We also presented results of the conducted experiments, which were carried out with limited sets of data and records from the UCI ML Repository, to evaluate the proposed framework. The results indicate that the proposed algorithm outperforms two conventional methods in terms of accuracy.

Further research will be to optimize the performance of the proposed algorithm by taking advantage of the new framework in a large scale application.

## References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. Proceedings of the 1993 ACM SIGMOD international conference on Management of data (1993) 207–216.
2. Amari, S. I.: Learning patterns and pattern sequences by self-organizing nets. IEEE Transactions on Computer, **21**(11) (1972) 1197–1206
3. Fayyad, U. M., Irani, K. B.: Multi-interval discretization of continuous-valued attributes for classification learning. IJCAI-93 (1993) 1022–1027
4. Hebb, D. O.: The Organization of Behaviour. John Wiley, New York, NY, (1949)

5. Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, **79**(8) (1982) 2554–2558
6. Kohavi, R., John, G., Long, R., Manley, D., Pfleger, K.: MLC++: a machine learning library in C++. *Tools with artificial intelligence* (1994) 740–743
7. Li, W., Han, J., Pei, J.: CMAR: Accurate and efficient classification based on Multiple Class-Association Rules, *IEEE*, (2001)
8. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In *KDD'98*, New York, NY, Aug. (1998)
9. Newman, D. J., Hettich, S., Blake, C. L., Merz, C. J.: UCI Repository of machine learning databases [<http://www.ics.uci.edu/mlearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science. (1998)
10. Quinlan, J. R.: C4.5: Programs for machine Learning. Morgan Kaufmann.



# Proposal of a New Conception of an Elastic Neural Network and Its Application to the Solution of a Two-Dimensional Travelling Salesman Problem

Tomasz Szatkiewicz

The Technical University of Koszalin, Department of Fine Mechanics,  
75-256 Koszalin, ul. Raławicka 15-17, Poland  
tszatkie@tu.koszalin.pl

**Abstract.** In this publication, a new conception of a neural network proposed by the author was described. It belongs to the class of self-organizing networks. The theory, structure and learning principles of the network proposed were all described. Further chapters include the application of a system of two proposed neural networks for the solution of a two-dimensional Euclides' travelling salesman problem. The feature of the neural network system proposed is an ability to avoid unfavourable local minima. The article presents graphical results of the evaluation of the neural network system from the initialization point to the determination of the selected TSP instance.

## 1 Introduction

Kohonen's papers [1], [2] resulted in the attempts to apply self-organizing neural networks to solve the problems of combinatorial optimization, and in particular the travelling salesman problem. The use of Kohonen's self-organizing network in the TSP case in question is that a neural network is constructed from neurons connected with each other into a ring structure. Each of them possesses two weights which constitute its  $x$  and  $y$  coordinates in the geometrical space. The learning (adaptation) of the network consists in providing on consecutive adaptation steps signal on the network input, which are two-dimensional vectors which depict points in the travelling salesman's task. The winning neuron and its neighbouring neurons are subject to adaptation. The adaptation strength is the greatest for the winning neuron and proportionally smaller for its neighbours. Interesting examples of the described technique were implemented by the following: Fritzsche [3], Kacalak [4], and Szatkiewicz [5].

Another interesting approach was an elastic neural network proposed by Durbin and Willshaw [6], in which neurons were combined with one another into a structure of a closed ring. An additional modification was an introduction of an elastic interaction between neighbouring neurons, which resulted in the occurrence of a factor minimizing the whole network length during the adaptation process. The adaptation technique itself was subject to change. All the vectors depicting points in the TSP task were given at the same time on the network input. All neurons on every learning stage were subject to adaptation, and the adaptation strength was proportional to the

geometrical distance of the point from the neuron and decreased with consecutive adaptation epochs.

The author proposed in his papers [7], [8] a modification of the elastic network presented by Durbin and Willshaw to increase the adaptation effectiveness. This modification consisted in the change of the neuron structure and its initialization in the form of neurons combined in a regular two-dimensional grid. Elastic interaction was simulated among the neurons. All the vectors which constitute the points in the TSP task were given on the network input, and the neural network itself reduced its structure during the adaptation process (it eliminated superfluous connections between neurons and superfluous neurons) to the form of a closed ring. Such an approach facilitated a search of a greater area of the states' space of the TSP task being solved and was the best from among all the abovementioned attempts as regards its quality.

Unfortunately, all the modifications described above did not prevent the network from generating of solutions which were unfavourable local minima of the states' space of the task being solved.

## 2 Structure of the Proposed Elastic Neuron Network

The proposed elastic neural network consists of  $N$  neurons, each of them possessing weight vector  $W_n = [w_x, w_y]^T$ , which represents its location in geometric space  $R^2$ . There are connections between the neurons. Each neuron is connected with two other neurons only, called the left neighbour and the right neighbour respectively, in such a way that the network consists, as regards its structure, a closed ring. Each connection between the neurons is characterised by parameter  $d_{n,n+1}$ , which constitutes the measure of the geometric distance between neighbouring neurons, in compliance with the accepted metric (Euclides' in this case) (Fig. 1).

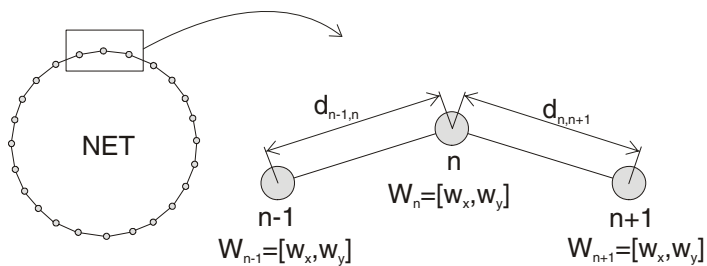


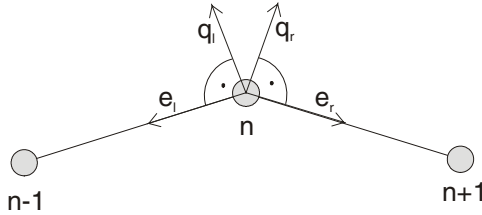
Fig. 1. Structure of an elastic neuron network in the form of a closed ring

## 3 Forces Acting on a Single Neuron During the Adaptation Process

Forces:  $q_l$ ,  $q_r$ ,  $e_l$  and  $e_r$  act on each neuron during the adaptation process. They result in the change of its weight vector  $W_n$ , and at the same time in the change of its

location in geometric space  $R^N$  (Fig. 2). These forces are components of the following two main classes of interactions:

- elastic interaction between a given neuron and its right and left neighbours,
- constant interaction, whose value is the function of network’s learning factor  $T_{NET}$ , hereinafter referred to as the network temperature.



**Fig. 2.** Forces acting on the neuron during the adaptation process of an elastic neuron network

### 4 Adaptation of an Elastic Neuron Network

The adaptation process of the proposed neuron network depends of the value change of parameter  $T$ , hereinafter referred to as the network temperature. It should be emphasized that in the network proposed, the value of  $T$  parameter increases during the adaptation process as it is the increase of  $T$  value which forces the network to search the up-to-date energetic minimum of its structure, with other parameters having been given.

The increment vector of the weights of a single neuron on a given adaptation step  $\Delta W_n(t)$  is the resultant vector of the four main component vectors in a given adaptation step:  $q_l(t)$ ,  $q_r(t)$ ,  $e_l(t)$  and  $e_r(t)$ .

$$\Delta W(t) = q_l(t) + q_r(t) + e_l(t) + e_r(t). \tag{1}$$

The vector change of the neuron’s weights:

$$W_n(t) = \Delta W_n(t) + W(t-1). \tag{2}$$

results in a displacement of a neuron in the space where network  $R^N$  is active.

#### 4.1 Determination of Constant Interaction Component Vectors

In order to calculate values  $|q_l|$  and  $|q_r|$ , first value  $Q$  has to be determined, which is described with the following formula:

$$Q = \frac{T}{V_{NET}}. \tag{3}$$

where:

$T$  – network temperature,

$V_{NET}$  – variable which represents network's geometric field in space

$$V_{NET} = \sum_{i=1}^{N-1} (n_{i_x} n_{i+1_y} - n_{i+1_x} n_{i_y}) \quad (4)$$

where:

$n_{i_x}$  and  $n_{i_y}$  – respective components x and y of weight vector of neuron  $n_i$ .

Next, values  $q_{n-1,n}$  and  $q_{n,n+1}$  are determined, which are necessary to determine values  $|q_l|$  and  $|q_r|$ . Adequately, from the following proportion:

$$\frac{Q}{q_{n-1,n}} = \frac{D_{NET}}{d_{n-1,n}} \quad \text{and} \quad \frac{Q}{q_{n,n+1}} = \frac{D_{NET}}{d_{n,n+1}} \quad (5)$$

where:

$d_{n-1,n}$  and  $d_{n,n+1}$  – geometric distances between neighbouring neurons interpreted as the lengths of connections between the left neighbour and the right neighbour respectively,

$D_{NET}$  – a sum of distances between connected (neighbouring) neurons in the network, interpreted as the total geometric length of the ring of neurons which constitutes the network's structure, determined with the following formula:

$$D_{NET} = \sum_{n=1}^{N-1} d_{n,n+1} + d_{n,1} \quad (6)$$

after rearranging equations (5) and suitable substitution, are obtained respectively:

$$q_{n,n+1} = \frac{Q d_{n,n+1}}{D_{NET}} \quad \text{and} \quad q_{n-1,n} = \frac{Q d_{n-1,n}}{D_{NET}} \quad (7)$$

the lengths of component vectors  $|q_l|$  and  $|q_r|$ . They are determined from the following formulas:

$$|q_l| = \frac{q_{n-1,n} d_{n-1,n}}{2} \quad \text{and} \quad |q_r| = \frac{q_{n,n+1} d_{n,n+1}}{2} \quad (8)$$

After suitable conversions and substitutions, the final formulas for the values of components  $|q_l|$  and  $|q_r|$  in the function of  $T$  temperature is obtained:

$$|q_l| = T \frac{d_{n-1,n}^2}{2D_{NET} V_{NET}} \quad \text{and} \quad |q_r| = T \frac{d_{n,n+1}^2}{2D_{NET} V_{NET}} \quad (9)$$

The direction of vectors  $q_l$  and  $q_r$  is perpendicular to the sections which join neuron  $n$  with respective neighbouring neurons  $n-1$  and  $n+1$ , while the sense of the vectors

can be depending of a specific application set in the outside or inside direction towards the geometrical form of the neuron network (Fig. 2).

#### 4.2 Determination of Component Vectors of Elastic Interaction

The lengths of component vectors of elastic interaction  $e_l$  and  $e_r$  are determined on the basis of the accepted elasticity function  $f_s(d, \kappa)$  and elasticity factor  $\kappa$ . A linear form of function  $f_s()$  was accepted, therefore values  $|e_l|$  and  $|e_r|$  are determined from the following formulae respectively:

$$|e_l| = \kappa \times d_{n-1,n} \quad \text{and} \quad |e_r| = \kappa \times d_{n,n+1}. \quad (10)$$

The direction of vectors  $e_l$  and  $e_r$  is parallel to the sections which join neuron  $n$  with respective neighbouring neurons  $n-1$  and  $n+1$  (Fig. 2).

### 5 Application of a System of Two Elastic Neural Networks for the Solution of the Travelling Salesman Problem

The elastic neural network described above can be successfully applied in a two-dimensional Euclides' travelling salesman problem. A characteristic feature of the elastic neuron network described is the fact that with a given value of the factor of network temperature  $T$ , the whole energy of the network  $E_{NET}$ , expressed with the following formula:

$$E_{NET} = \sum_{i=1}^N (|e_{l_i}| + |e_{r_i}|). \quad (11)$$

in accordance with Hamilton's energy minimization, always approaches a minimum, and because  $E_{NET}$  is proportional to total network length  $D_{NET}$ , one can put forward a thesis that the network during the adaptation tends to preserve its minimum total length.

In order to apply an elastic neuron network for the solution of a two-dimensional travelling salesman problem, the two described neuron network should be initiated respectively in space  $R^2$  of a given instance of the TSP problem with the following initial conditions:

1. One network (internal – hereinafter referred to as IENN) is totally included inside the other one (external – hereinafter referred to as EENN),
2. No point from the set of nodes of a given instance of the travelling salesman problem is located inside IENN,
3. No point from the set of nodes of a given instance of the travelling salesman problem is located outside EENN.

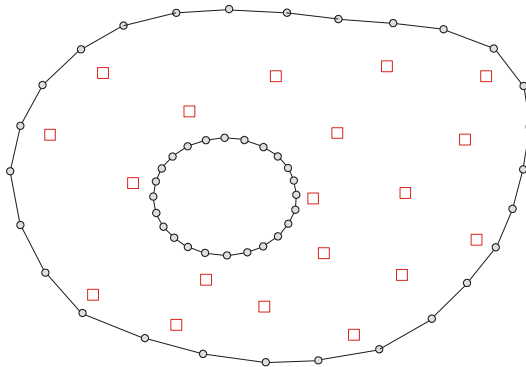
A graphic form of the system of two elastic neuron networks after the initiation is represented in Fig. 3.

The initiation of the system occurs in accordance with the following conditions:  
 For EENN, the value of Q is determined with the following formula:

$$Q = -\frac{T}{V_{NET}} \tag{12}$$

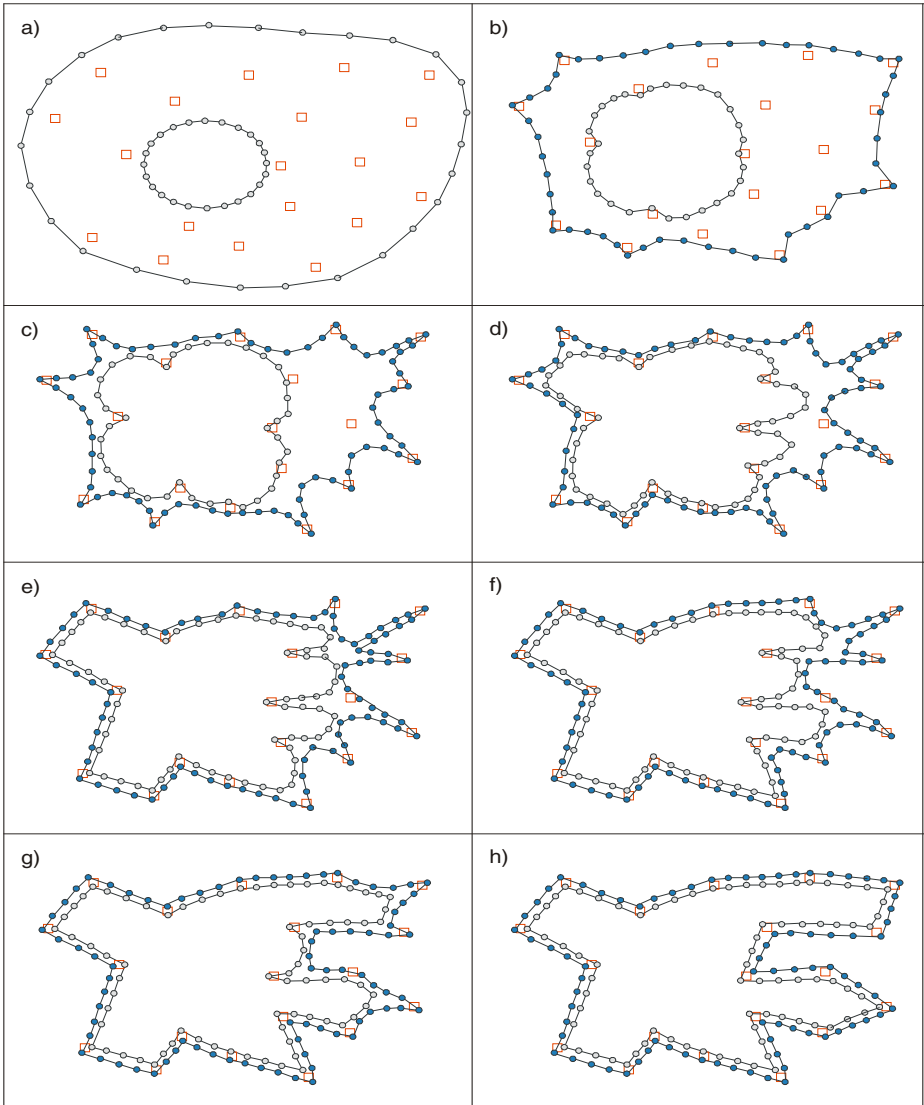
that is it has an opposite sign than for the value of Q for IENN,

1. No nod of the instance of the travelling salesman problem being solved can be found inside IENN,
2. No nod of the instance of the travelling salesman problem being solved can be found outside EENN,
3. No part of EENN (no neuron) can be found inside IENN,
4. No part of IENN (no neuron) can be found outside EENN,
5. Between the neurons of EENN and IENN, there is no interaction, and in particular, no friction,
6. Between the neurons of EENN and IENN and the nods of the instance of the travelling salesman problem being solved there is no interaction, and in particular, no friction.



**Fig. 3.** System of two elastic neural networks for travelling salesman problem after inicialization

The adaptation process of the described system of elastic neuron networks consists in a gradual increase of the values of T temperature factor of both networks of EENN and IENN. Both neuron networks during adaptation come across obstacles n the form of points in the TSP instance being solved and in the form of their own neurons. Assuming that factors T of both networks are equal, on the basis of Hamilton’s energy minimization, one can put forward a thesis that such a system of neural networks, with a sufficiently high level of T parameter, will stabilize in such a point where the surfaces of both networks will adhere to each other on their whole length, and all the points of the TSP instance being solved will neighbour with the neurons of both neuron networks of EENN and IENN. A geometric picture of the system of two elastic neural networks after an adaptation with as system of the points of the TSP instance being solved constitutes the solution of a given TSP instance (Fig. 4h). Such a system has an astonishing ability of by-passing local minima.



**Fig. 4.** Graphic picture of adaptation of system of elastic neuron networks for the solution of travelling salesman problem

## 6 Other Possible Applications of the Elastic Neuron Network

The elastic neuron network described can be applied for various issues in the field of system optimization, recognition of standards, issues of packing, cutting, determination of influence zones, and many others. There are many possibilities of modifications to

increase the possible application areas of the elastic neuron network. The main areas include the following:

- an increase of the number of the dimensions of the working area of the network,
- a differentiation of the functions of elastic interaction between neighbouring neurons in such a manner that the network is not homogenous,
- introduction of additional signals which affect the network during the adaptation process,
- an introduction of a possibility of a dynamic change of the network structure during the adaptation process,
- a modification of the network's learning principles.

The ability of the elastic neuron network to by-pass unfavourable local minima and its ability of an adaptation and a structural self-configuration, with a simultaneous continuous tendency of the network to remain in an energy minimum, creates special possibilities of the applications of the elastic network described in the optimization area in the theory of systems.

## References

1. Kohonen T.: Selforganization and associative memory. Springer Series in Information Science. Springer, Berlin Heidelberg, New York 2<sup>nd</sup> edition (1988)
2. Kohonen T.: The Self-Organizing Map. Proceedings of the IEEE, special issue on Neural Networks, vol. 78 (1990)
3. Fritzke B, Wilke P.: FLEXMAP – neural network with linear time and space complexity for the travelling salesman problem. Proc. Int. Joint Conference of Neural Networks pp 929-934 Singapore (1991)
4. Kacalak W., Wawryn K.: Some aspects of the modified competitive self learning neural network algorithm. Int. Conf. Artificial Intelligence Networks in Engineering ANNIE'94, St. Louis, USA, ASME Press (1994)
5. Szatkiewicz T., Kacalak W.: New algorithms for trajectory optimisation of tools and objects in production systems. II Int. Conf. CANT, Wrocław (2003)
6. Durbin R., Willshaw D.: An Analogue Approach for the Travelling Salesman Problem Using an Elastic Net Method. Nature, vol. 326, 644-647 (1987)
7. Szatkiewicz T.: Adaptation of two-dimensional elastic neural net for travelling salesman problem. Polioptimization and CAD, Mielno (2004),
8. Szatkiewicz T.: Hybrid, self-adaptive optimisation system of movement of tools and objects in production systems. PhD Thesis, Technical University Of Koszalin Faculty of Mechanics, Koszalin (2005)



# Robust Stability Analysis for Delayed BAM Neural Networks

Yijing Wang and Zhiqiang Zuo\*

School of Electrical Engineering & Automation, Tianjin University,  
Tianjin, 300072, China  
{yjiang, zqzuo}@tju.edu.cn

**Abstract.** The problem of robust stability for a class of uncertain bidirectional associative memory neural networks with time delays is investigated in this paper. A more general Lyapunov-Krasovskii functional is proposed to derive a less conservative robust stability condition within the framework of linear matrix inequalities. A numerical example is given to illustrate the effectiveness of the proposed method.

## 1 Introduction

Since Kosko proposed the model of a class of two-layer heteroassociative networks called bidirectional associative memory (BAM) neural networks [1], the problems of existence of the equilibrium points, bounds of trajectories and stability of BAM neural networks have attracted the interest of many researchers. There are many applications for BAM neural networks such as pattern recognition, artificial intelligence and automatic control. In the electronic implementation of analog neural networks, time delays occur in the communication and response of neurons owing to the finite switching speed of amplifier. It is well known that time delay can influence the stability of a network by creating oscillatory or unstable phenomena. Therefore, the study of BAM neural networks with consideration of time delays has received considerable attention for a long time (see e.g., [2]-[8] and the references therein). Recently, Park [9] studied the robust stability problem for delayed bidirectional associative memory neural networks with norm bounded uncertainties. Two mathematical operators were introduced to transform the original neural network. Furthermore, an integral inequality was adopted to derive the stability condition. As we know, model transformation or bounding method may bring conservatism in stability analysis.

To reduce the conservatism in [9], we construct a new type of Lyapunov-Krasovskii functional and a less conservative result is obtained by using some free matrices to express the relationship among the terms in dynamical equation of the neural network. Neither model transformation nor bounding technique on cross term is used. It should be noted that although the stability condition is

---

\* Corresponding author.

delay-independent, its superiority over the existing method [9] can be shown in Section 4.

**Notations:** The following notations are used throughout this paper.  $\mathfrak{R}$  is the set of real numbers.  $\mathfrak{R}^n$ ,  $\mathfrak{R}^{m \times n}$  are sets of real vectors with dimension  $n$  and real matrices with dimension  $m \times n$ , respectively. The notation  $X \geq Y$  (respectively,  $X > Y$ ), where  $X$  and  $Y$  are symmetric matrices, means that the matrix  $X - Y$  is positive semi-definite (respectively, positive definite). In symmetric block matrices, we use  $\star$  to denote terms that are induced by symmetry. For simplicity, we use  $G + (*)$  to represent  $G + G^T$ .

## 2 Problems Statement and Preliminaries

Consider the following delayed BAM neural network in [9]

$$\begin{aligned} \dot{u}_i(t) &= -(a_i + \Delta a_i)u_i(t) + \sum_{j=1}^m (w_{ji} + \Delta w_{ji})g_j(v_j(t - \tau)) + I_i, \quad i = 1, 2, \dots, n \\ \dot{v}_j(t) &= -(b_j + \Delta b_j)v_j(t) + \sum_{i=1}^n (v_{ij} + \Delta v_{ij})f_i(u_i(t - \sigma)) + J_j, \quad j = 1, 2, \dots, m \end{aligned} \quad (1)$$

or

$$\begin{aligned} \dot{u}(t) &= -(A + \Delta A)u(t) + (W + \Delta W)^T g(v(t - \tau)) + I \\ \dot{v}(t) &= -(B + \Delta B)v(t) + (V + \Delta V)^T f(u(t - \sigma)) + J \end{aligned} \quad (2)$$

where  $u_i$  and  $v_j$  are the activations of the  $i$ th neuron and the  $j$ th neuron, respectively.  $w_{ji}$  and  $v_{ij}$  are the connection weights.  $I_i$  and  $J_j$  denote the external inputs.  $\tau > 0$  and  $\sigma > 0$  correspond to the finite speed of the axonal signal transmission delays.  $\Delta A = (\Delta a_i)_{n \times n}$ ,  $\Delta B = (\Delta b_j)_{m \times m}$ ,  $\Delta W = (\Delta w_{ji})_{m \times n}$ , and  $\Delta V = (\Delta v_{ij})_{n \times m}$  are uncertain matrices which are defined as follows

$$\begin{aligned} \Delta A &= H_1 F_1(t) E_1, & \Delta B &= H_2 F_2(t) E_2 \\ \Delta W &= H_3 F_3(t) E_3, & \Delta V &= H_4 F_4(t) E_4 \end{aligned} \quad (3)$$

where  $H_i$ ,  $E_i$ , ( $i = 1, 2, 3, 4$ ) are constant matrices and  $F_i(t)$ , ( $i = 1, 2, 3, 4$ ) are unknown real time-varying matrices satisfying  $F_i^T(t)F_i(t) \leq I$ .

Throughout this paper, we make the following assumptions.

**Assumption 1.**  $g_j$  and  $f_i$  are bounded on  $\mathfrak{R}$ .

**Assumption 2.** There exist positive numbers  $k_j > 0$  and  $l_i > 0$  such that

$$0 < \frac{g_j(\zeta_1) - g_j(\zeta_2)}{\zeta_1 - \zeta_2} \leq k_j, \quad 0 < \frac{f_i(\zeta_1) - f_i(\zeta_2)}{\zeta_1 - \zeta_2} \leq l_i$$

for all  $\zeta_1, \zeta_2 \in \mathfrak{R}$ ,  $\zeta_1 \neq \zeta_2$ .

As we know, neural network (II) has at least one equilibrium point under Assumption 1 and 2.

**Definition 1.** *Bidirectional associative memory neural network (I) with time delay is said to be robustly stable if there exists a unique equilibrium point  $[u^* v^*]^T$  of the system, which is asymptotically stable in the presence of the parameter uncertainties  $\Delta A, \Delta B, \Delta W$  and  $\Delta V$ .*

If we introduce the transformation  $x(t) = u(t) - u^*, y(t) = v(t) - v^*$ , neural network (II) can be transformed to

$$\begin{aligned} \dot{x}(t) &= -(A + \Delta A)x(t) + (W + \Delta W)^T g_1(y(t - \tau)) \\ \dot{y}(t) &= -(B + \Delta B)y(t) + (V + \Delta V)^T f_1(x(t - \sigma)) \end{aligned} \tag{4}$$

where  $g_{1j}(y_j(t - \tau)) = g_j(y_j(t - \tau) + v_j^*) - g_j(v_j^*)$  and  $f_{1i}(x_i(t - \sigma)) = f_i(x_i(t - \sigma) + u_i^*) - f_i(u_i^*)$ . It is easily verified that

$$0 \leq \frac{g_{1j}(z_j)}{z_j} \leq k_j, \quad 0 \leq \frac{f_{1i}(z_i)}{z_i} \leq l_i, \quad \forall z_j, z_i \neq 0 \tag{5}$$

Obviously, the robust stability of the equilibrium point for neural network (II) is equivalent to the robust stability of the origin for system (4).

By (5), we have

$$2 \sum_{i=1}^n \gamma_i f_{1i}(x_i(t - \sigma)) [l_i x_i(t - \sigma) - f_{1i}(x_i(t - \sigma))] \geq 0 \tag{6}$$

$$2 \sum_{j=1}^m s_j g_{1j}(y_j(t - \tau)) [k_j y_j(t - \tau) - g_{1j}(y_j(t - \tau))] \geq 0 \tag{7}$$

for any scalars  $\gamma_i \geq 0, s_j \geq 0, i = 1, 2, \dots, n, j = 1, 2, \dots, m$ , which can be rewritten as

$$2f_1^T(x(t - \sigma))\Gamma Lx(t - \sigma) - 2f_1^T(x(t - \sigma))\Gamma f_1(x(t - \sigma)) \geq 0 \tag{8}$$

$$2g_1^T(y(t - \tau))SKy(t - \tau) - 2g_1^T(y(t - \tau))Sg_1(y(t - \tau)) \geq 0 \tag{9}$$

where

$$\Gamma = \text{diag}\{\gamma_1, \gamma_2, \dots, \gamma_n\} \geq 0$$

$$S = \text{diag}\{s_1, s_2, \dots, s_m\} \geq 0$$

$$L = \text{diag}\{l_1, l_2, \dots, l_n\} > 0$$

$$K = \text{diag}\{k_1, k_2, \dots, k_m\} > 0$$

In order to obtain our main results in this paper, the following lemma is needed which is used frequently to deal with the norm-bounded uncertainties such as [10].

**Lemma 1.** Given matrices  $Q = Q^T$ ,  $H$  and  $E$ , the matrix inequality

$$Q + HFE + E^T F^T H^T < 0$$

holds for all  $F$  satisfying  $F^T F \leq I$  if and only if there exists a scalar  $\varepsilon > 0$  such that

$$Q + \varepsilon HH^T + \varepsilon^{-1} E^T E < 0$$

### 3 Main Results

Firstly, we consider the delayed bidirectional associative memory neural network (4) without uncertainties, i.e.,

$$\begin{aligned} \dot{x}(t) &= -Ax(t) + W^T g_1(y(t - \tau)) \\ \dot{y}(t) &= -By(t) + V^T f_1(x(t - \sigma)) \end{aligned} \quad (10)$$

**Theorem 1.** The nominal delayed bidirectional associative memory neural network described by (10) is stable if there exist matrices  $P_1 > 0$ ,  $P_2 > 0$ ,  $\Gamma = \text{diag}\{\gamma_1, \gamma_2, \dots, \gamma_m\} \geq 0$ ,  $S = \text{diag}\{s_1, s_2, \dots, s_m\} \geq 0$ ,  $N_i, T_i$  ( $i = 1, \dots, 10$ ),  $Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \geq 0$ ,  $R = \begin{bmatrix} R_{11} & R_{12} \\ R_{12}^T & R_{22} \end{bmatrix} \geq 0$  such that

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} & \Sigma_{14} & \Sigma_{15} & \Sigma_{16} & \Sigma_{17} & \Sigma_{18} & \Sigma_{19} & \Sigma_{1,10} \\ * & \Sigma_{22} & \Sigma_{23} & \Sigma_{24} & \Sigma_{25} & \Sigma_{26} & 0 & \Sigma_{28} & \Sigma_{29} & 0 \\ * & * & \Sigma_{33} & \Sigma_{34} & \Sigma_{35} & \Sigma_{36} & \Sigma_{37} & \Sigma_{38} & \Sigma_{39} & \Sigma_{3,10} \\ * & * & * & \Sigma_{44} & \Sigma_{45} & \Sigma_{46} & \Sigma_{47} & \Sigma_{48} & \Sigma_{49} & \Sigma_{4,10} \\ * & * & * & * & \Sigma_{55} & \Sigma_{56} & 0 & \Sigma_{58} & \Sigma_{59} & 0 \\ * & * & * & * & * & \Sigma_{66} & \Sigma_{67} & \Sigma_{68} & \Sigma_{69} & \Sigma_{6,10} \\ * & * & * & * & * & * & \Sigma_{77} & \Sigma_{78} & \Sigma_{79} & \Sigma_{7,10} \\ * & * & * & * & * & * & * & \Sigma_{88} & \Sigma_{89} & \Sigma_{8,10} \\ * & * & * & * & * & * & * & * & \Sigma_{99} & \Sigma_{9,10} \\ * & * & * & * & * & * & * & * & * & \Sigma_{10,10} \end{bmatrix} < 0 \quad (11)$$

where

$$\begin{aligned} \Sigma_{11} &= N_1 A + A^T N_1^T + Q_{11} \\ \Sigma_{12} &= A^T N_2^T \\ \Sigma_{13} &= A^T N_3^T - T_1 V^T \\ \Sigma_{14} &= N_1 + A^T N_4^T + P_1 + Q_{12} \\ \Sigma_{15} &= A^T N_5^T \\ \Sigma_{16} &= A^T N_6^T + T_1 B \\ \Sigma_{17} &= A^T N_7^T \\ \Sigma_{18} &= A^T N_8^T - N_1 W^T \end{aligned}$$

$$\begin{aligned}
\Sigma_{19} &= A^T N_9^T + T_1 \\
\Sigma_{1,10} &= A^T N_{10}^T \\
\Sigma_{22} &= -Q_{11} \\
\Sigma_{23} &= -T_2 V^T + LR \\
\Sigma_{24} &= N_2 \\
\Sigma_{25} &= -Q_{12} \\
\Sigma_{26} &= T_2 B \\
\Sigma_{28} &= -N_2 W^T \\
\Sigma_{29} &= T_2 \\
\Sigma_{33} &= -T_3 V^T - VT_3^T - 2R \\
\Sigma_{34} &= N_3 - VT_4^T \\
\Sigma_{35} &= -VT_5^T \\
\Sigma_{36} &= T_3 B - VT_6^T \\
\Sigma_{37} &= -VT_7^T \\
\Sigma_{38} &= -N_3 W^T - VT_8^T \\
\Sigma_{39} &= T_3 - VT_9^T \\
\Sigma_{3,10} &= -VT_{10}^T \\
\Sigma_{44} &= N_4 + N_4^T + Q_{22} \\
\Sigma_{45} &= N_5^T \\
\Sigma_{46} &= T_4 B + N_6^T \\
\Sigma_{47} &= N_7^T \\
\Sigma_{48} &= N_8^T - N_4 W^T \\
\Sigma_{49} &= N_9^T + T_4 \\
\Sigma_{4,10} &= N_{10}^T \\
\Sigma_{55} &= -Q_{22} \\
\Sigma_{56} &= T_5 B \\
\Sigma_{58} &= -N_5 W^T \\
\Sigma_{59} &= T_5 \\
\Sigma_{66} &= T_6 B + B^T T_6^T + R_{11} \\
\Sigma_{67} &= B^T T_7^T \\
\Sigma_{68} &= -N_6 W^T + B^T T_8^T
\end{aligned}$$

$$\begin{aligned}
\Sigma_{69} &= B^T T_9^T + P_2 + T_6 + R_{12} \\
\Sigma_{6,10} &= B^T T_{10}^T \\
\Sigma_{77} &= -R_{11} \\
\Sigma_{78} &= -N_7 W^T + K S \\
\Sigma_{79} &= T_7 \\
\Sigma_{7,10} &= -R_{12} \\
\Sigma_{88} &= -N_8 W^T - W N_8^T - 2S \\
\Sigma_{89} &= -W N_9^T + T_8 \\
\Sigma_{8,10} &= -W N_{10}^T \\
\Sigma_{99} &= T_9 + T_9^T + R_{22} \\
\Sigma_{9,10} &= T_{10}^T \\
\Sigma_{10,10} &= -R_{22}
\end{aligned}$$

*Proof.* Choose the Lyapunov-Krasovskii functional as

$$\begin{aligned}
V(t) &= x^T(t) P_1 x(t) + y^T(t) P_2 y(t) + \int_{t-\sigma}^t \begin{bmatrix} x(s) \\ \dot{x}(s) \end{bmatrix}^T \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \begin{bmatrix} x(s) \\ \dot{x}(s) \end{bmatrix} ds \\
&\quad + \int_{t-\tau}^t \begin{bmatrix} y(s) \\ \dot{y}(s) \end{bmatrix}^T \begin{bmatrix} R_{11} & R_{12} \\ R_{12}^T & R_{22} \end{bmatrix} \begin{bmatrix} y(s) \\ \dot{y}(s) \end{bmatrix} ds
\end{aligned} \tag{12}$$

For any compatible dimensioned matrices  $N_i, T_i$  ( $i = 1, 2, \dots, 10$ ), we have

$$\begin{aligned}
\alpha &= 2 [x^T(t) N_1 + x^T(t-\sigma) N_2 + f_1^T(x(t-\sigma)) N_3 + \dot{x}^T(t) N_4 + \dot{x}^T(t-\sigma) N_5 \\
&\quad + y^T(t) N_6 + y^T(t-\tau) N_7 + g_1^T(y(t-\tau)) N_8 + \dot{y}^T(t) N_9 + \dot{y}^T(t-\tau) N_{10}] \\
&\quad \times [\dot{x}(t) + Ax(t) - W^T g_1(y(t-\tau))] \equiv 0
\end{aligned} \tag{13}$$

$$\begin{aligned}
\beta &= 2 [x^T(t) T_1 + x^T(t-\sigma) T_2 + f_1^T(x(t-\sigma)) T_3 + \dot{x}^T(t) T_4 + \dot{x}^T(t-\sigma) T_5 \\
&\quad + y^T(t) T_6 + y^T(t-\tau) T_7 + g_1^T(y(t-\tau)) T_8 + \dot{y}^T(t) T_9 + \dot{y}^T(t-\tau) T_{10}] \\
&\quad \times [\dot{y}(t) + By(t) - V^T f_1(x(t-\sigma))] \equiv 0
\end{aligned} \tag{14}$$

The time derivative of  $V(t)$  along the trajectory of neural network (10) is

$$\begin{aligned}
\dot{V}(t) &\leq 2x^T(t) P_1 \dot{x}(t) + 2y^T(t) P_2 \dot{y}(t) + \alpha + \beta + \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix}^T \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} \\
&\quad - \begin{bmatrix} x(t-\sigma) \\ \dot{x}(t-\sigma) \end{bmatrix}^T \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \begin{bmatrix} x(t-\sigma) \\ \dot{x}(t-\sigma) \end{bmatrix} + \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}^T \begin{bmatrix} R_{11} & R_{12} \\ R_{12}^T & R_{22} \end{bmatrix} \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix} \\
&\quad - \begin{bmatrix} y(t-\tau) \\ \dot{y}(t-\tau) \end{bmatrix}^T \begin{bmatrix} R_{11} & R_{12} \\ R_{12}^T & R_{22} \end{bmatrix} \begin{bmatrix} y(t-\tau) \\ \dot{y}(t-\tau) \end{bmatrix} + 2f_1^T(x(t-\sigma)) \Gamma L x(t-\sigma) \\
&\quad - 2f_1^T(x(t-\sigma)) \Gamma f_1(x(t-\sigma)) + 2g_1^T(y(t-\tau)) S K y(t-\tau) \\
&\quad - 2g_1^T(y(t-\tau)) S g_1(y(t-\tau)) \\
&= \xi^T(t) \Sigma \xi(t)
\end{aligned} \tag{15}$$

where

$$\xi^T(t) = [x^T(t) \ x^T(t - \sigma) \ f_1^T(x(t - \sigma)) \ \dot{x}^T(t) \ \dot{x}^T(t - \sigma) \\ y^T(t) \ y^T(t - \tau) \ g_1^T(y(t - \tau)) \ \dot{y}^T(t) \ \dot{y}^T(t - \tau)]^T$$

By [12], if  $\Sigma < 0$ , then  $\dot{V}(t) < 0$  for any  $\xi(t) \neq 0$ , which guarantees the asymptotically stable for delayed bidirectional associative memory neural network (10). This completes the proof.

*Remark 1.* Theorem 1 provides a new stability criterion for nominal bidirectional associative memory neural network with time delay (10). Unlike the method proposed in [9], we do not use any model transformation or integral inequality in this paper, which ensures the less conservatism of Theorem 1. Some free weighting matrices are introduced to bring some flexibility in solving LMI.

Based on the result of Theorem 1, it is easy to obtain the robust stability condition for delayed bidirectional associative memory neural network (11) with uncertainties  $\Delta A, \Delta B, \Delta W, \Delta V$ .

**Theorem 2.** *The uncertain bidirectional associative memory neural network with time delay (11) is robustly stable if there exist matrices  $P_1 > 0, P_2 > 0$ , diagonal matrices  $R \geq 0, S \geq 0, N_i, T_i (i = 1, \dots, 10), Q \geq 0, R \geq 0$ , and positive scalars  $\varepsilon_i > 0 (i = 1, 2, 3, 4)$  such that*

$$\begin{bmatrix} \bar{\Sigma}_{11} & \Sigma_{12} & \Sigma_{13} & \Sigma_{14} & \Sigma_{15} & \Sigma_{16} & \Sigma_{17} & \Sigma_{18} & \Sigma_{19} & \Sigma_{1,10} & N_1 H_1 & T_1 H_2 & -N_1 E_3^T & -T_1 E_4^T \\ * & \Sigma_{22} & \Sigma_{23} & \Sigma_{24} & \Sigma_{25} & \Sigma_{26} & 0 & \Sigma_{28} & \Sigma_{29} & 0 & N_2 H_1 & T_2 H_2 & -N_2 E_3^T & -T_2 E_4^T \\ * & * & \bar{\Sigma}_{33} & \Sigma_{34} & \Sigma_{35} & \Sigma_{36} & \Sigma_{37} & \Sigma_{38} & \Sigma_{39} & \Sigma_{3,10} & N_3 H_1 & T_3 H_2 & -N_3 E_3^T & -T_3 E_4^T \\ * & * & * & \Sigma_{44} & \Sigma_{45} & \Sigma_{46} & \Sigma_{47} & \Sigma_{48} & \Sigma_{49} & \Sigma_{4,10} & N_4 H_1 & T_4 H_2 & -N_4 E_3^T & -T_4 E_4^T \\ * & * & * & * & \Sigma_{55} & \Sigma_{56} & 0 & \Sigma_{58} & \Sigma_{59} & 0 & N_5 H_1 & T_5 H_2 & -N_5 E_3^T & -T_5 E_4^T \\ * & * & * & * & * & \bar{\Sigma}_{66} & \Sigma_{67} & \Sigma_{68} & \Sigma_{69} & \Sigma_{6,10} & N_6 H_1 & T_6 H_2 & -N_6 E_3^T & -T_6 E_4^T \\ * & * & * & * & * & * & \Sigma_{77} & \Sigma_{78} & \Sigma_{79} & \Sigma_{7,10} & N_7 H_1 & T_7 H_2 & -N_7 E_3^T & -T_7 E_4^T \\ * & * & * & * & * & * & * & \bar{\Sigma}_{88} & \Sigma_{89} & \Sigma_{8,10} & N_8 H_1 & T_8 H_2 & -N_8 E_3^T & -T_8 E_4^T \\ * & * & * & * & * & * & * & * & \Sigma_{99} & \Sigma_{9,10} & N_9 H_1 & T_9 H_2 & -N_9 E_3^T & -T_9 E_4^T \\ * & * & * & * & * & * & * & * & * & \Sigma_{10,10} & N_{10} H_1 & T_{10} H_2 & -N_{10} E_3^T & -T_{10} E_4^T \\ * & * & * & * & * & * & * & * & * & * & -\varepsilon_1 I & 0 & 0 & 0 \\ * & * & * & * & * & * & * & * & * & * & * & -\varepsilon_2 I & 0 & 0 \\ * & * & * & * & * & * & * & * & * & * & * & * & -\varepsilon_3 I & 0 \\ * & * & * & * & * & * & * & * & * & * & * & * & * & -\varepsilon_4 I \end{bmatrix} < 0 \tag{16}$$

where

$$\begin{aligned} \bar{\Sigma}_{11} &= N_1 A + A^T N_1^T + Q_{11} + \varepsilon_1 E_1^T E_1 \\ \bar{\Sigma}_{33} &= -T_3 V^T - V T_3^T - 2R + \varepsilon_4 H_4 H_4^T \\ \bar{\Sigma}_{66} &= T_6 B + B^T T_6^T + R_{11} + \varepsilon_2 E_2^T E_2 \\ \bar{\Sigma}_{88} &= -N_8 W^T - W N_8^T - 2S + \varepsilon_3 H_3 H_3^T \end{aligned}$$

*Proof.* Replacing  $A, B, W$  and  $V$  in (11) with  $A + H_1 F_1(t) E_1, B + H_2 F_2(t) E_2, W + H_3 F_3(t) E_3$  and  $V + H_4 F_4(t) E_4$ , we can easily obtain that uncertain delayed BAM neural network (11) is robustly stable if the following condition holds

$$\begin{aligned} &\Sigma + M_1 F_1(t) G_1^T + (*) + M_2 F_2(t) G_2^T + (*) \\ &+ M_3 F_3^T(t) G_3^T + (*) + M_4 F_4^T(t) G_4^T + (*) < 0 \end{aligned}$$

where

$$M_1 = \begin{bmatrix} N_1 H_1 \\ N_2 H_1 \\ N_3 H_1 \\ N_4 H_1 \\ N_5 H_1 \\ N_6 H_1 \\ N_7 H_1 \\ N_8 H_1 \\ N_9 H_1 \\ N_{10} H_1 \end{bmatrix}, M_2 = \begin{bmatrix} T_1 H_2 \\ T_2 H_2 \\ T_3 H_2 \\ T_4 H_2 \\ T_5 H_2 \\ T_6 H_2 \\ T_7 H_2 \\ T_8 H_2 \\ T_9 H_2 \\ T_{10} H_2 \end{bmatrix}, M_3 = \begin{bmatrix} -N_1 E_3^T \\ -N_2 E_3^T \\ -N_3 E_3^T \\ -N_4 E_3^T \\ -N_5 E_3^T \\ -N_6 E_3^T \\ -N_7 E_3^T \\ -N_8 E_3^T \\ -N_9 E_3^T \\ -N_{10} E_3^T \end{bmatrix}, M_4 = \begin{bmatrix} -T_1 E_4^T \\ -T_2 E_4^T \\ -T_3 E_4^T \\ -T_4 E_4^T \\ -T_5 E_4^T \\ -T_6 E_4^T \\ -T_7 E_4^T \\ -T_8 E_4^T \\ -T_9 E_4^T \\ -T_{10} E_4^T \end{bmatrix}$$

$$G_1^T = [E_1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], \quad G_2^T = [0 \ 0 \ 0 \ 0 \ 0 \ E_2 \ 0 \ 0 \ 0 \ 0]$$

$$G_3^T = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ H_3^T \ 0 \ 0], \quad G_4^T = [0 \ 0 \ H_4^T \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

By Lemma 1, (16) follows directly.

*Remark 2.* The method proposed in this paper can be easily extended to obtain the robust stability condition for uncertain systems with linear fractional form studied in [13] which is more general than the norm-bounded ones.

## 4 Example

In order to demonstrate the effectiveness of the method we have presented, an example is given in this section to compare with the results of the previous methods.

Consider the same delayed bidirectional associative memory neural network (4) with norm-bounded uncertainties studied in [9]

$$A = \begin{bmatrix} 2.2 & 0 \\ 0 & 1.3 \end{bmatrix}, \quad B = \begin{bmatrix} 1.2 & 0 \\ 0 & 1.1 \end{bmatrix}$$

$$W = \begin{bmatrix} 0.1 & 0.15 \\ 0.2 & 0.1 \end{bmatrix}, \quad V = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.3 \end{bmatrix}$$

$$H_1 = H_2 = H_3 = H_4 = I$$

$$E_1 = E_2 = -0.2I, \quad E_3 = E_4 = 0.2I, \quad \sigma = \tau$$

By choosing  $f_i(x) = 0.5[|x_i + 1| - |x_i - 1|]$ ,  $g_j(y) = 0.5[|y_j + 1| - |y_j - 1|]$ , we have  $K = L = I$ . Furthermore, we assume that

$$F_1(t) = F_3(t) = \begin{bmatrix} \sin(t) & 0 \\ 0 & \cos(t) \end{bmatrix}, \quad F_2(t) = F_4(t) = \begin{bmatrix} \cos(t) & 0 \\ 0 & \sin(t) \end{bmatrix} \quad (17)$$

Using the Matlab LMI-Toolbox to solve this problem, we find that the robust stability condition proposed in Theorem 2 is feasible, which means that



this uncertain delayed bidirectional associative memory neural networks delay-independent stable (that is, the neural network is robustly stable no matter what the values of  $\sigma$  and  $\tau$  are). However, the maximal bound of delay  $\sigma$  is dependent on the introduced parameters  $\gamma_1$  and  $\gamma_2$  in Theorem 1 of [9]. When  $\gamma_1 = \gamma_2 = 2$ , the maximal bound of delay  $\sigma$  is 0.5232. It is obvious that the stability criterion presented in this paper gives a less conservative result than that in [9].

For given initial condition  $[-1 \ 0.5 \ 1 \ -0.5]^T$ , we obtain a numerical simulation result by choosing  $\sigma = \tau = 30$ . Its convergence behavior is shown in Figure 1. As we can see, the state of this delayed bidirectional associative memory neural network is indeed robustly stable.

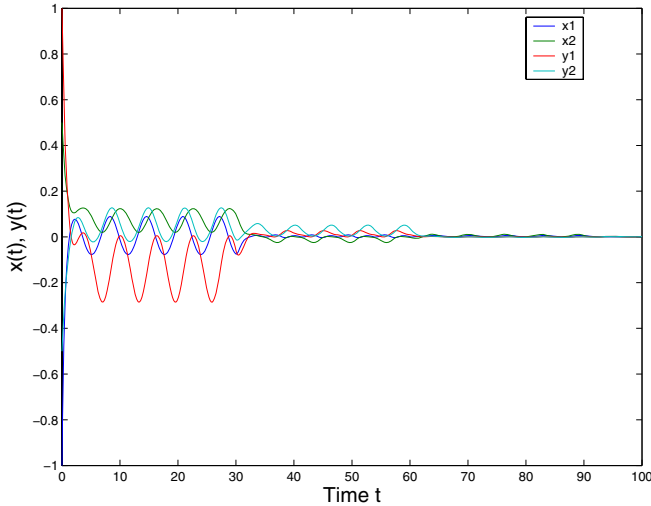


Fig. 1. The convergence dynamics of the neural network in Example 1

## 5 Conclusion

In this paper, we have studied the delay-independent robust stability for bidirectional associative memory neural networks subject to both time delay and norm-bounded uncertainties. Based on a new Lyapunov-Krasovskii functional and the introduction of free weighting matrices, a less conservative robust stability criterion has been derived in terms of linear matrix inequalities which is computationally efficient. An example has shown that our method gives an improvement over the existing ones.

**Acknowledgments.** This work is supported by National Natural Science Foundation of China (No. 60504011) and (No. 60504012).

## References

1. B. Kosko, Bi-directional associative memories, *IEEE Trans. Syst. Man Cybernet.*, vol. 18, pp. 49-60, 1988.
2. K. Gopalsamy and X. Z. He, Delay-independent stability in bidirectional associative memory networks, *IEEE Trans. Neural Networks*, vol. 5, no. 6, pp. 998-1002, Nov. 1994.
3. J. Cao, Global asymptotic stability of delayed bi-directional associative memory neural networks, *Applied Mathematics and Computation*, vol. 142, pp. 333-339, 2003.
4. A. Chen, J. Cao, and L. Huang, An estimation of upperbound of delays for global asymptotic stability of delayed Hopfield neural networks, *IEEE Tran. Circuit Syst. I*, vol. 49, no. 7, pp. 1028-1032, Jul. 2002.
5. Y. K. Li, Global exponential stability of BAM neural networks with delays and impulses, *Chaos Solitons Fractals*, vol. 24, pp. 279-285, 2005.
6. C. D. Li, X. F. Liao, and R. Zhang, Delay-dependent exponential stability analysis of bi-directional associative memory neural networks with time delay: an LMI approach, *Chaos, Solitons and Fractals*, vol. 24, pp. 1119-1134, 2005.
7. X. Hunag, J. Cao, and D. Huang, LMI-based approach for delay-dependent exponential stability analysis of BAM neural networks, *Chaos, Solitons and Fractals*, vol. 24, pp. 885-898. 2005.
8. Y. R. Liu, Z. Wang and X. H. Liu, Global asymptotic stability of generalized bi-directional associative memory networks with discrete and distributed delays, *Chaos, Solitons and Fractals*, vol. 28, pp. 793-803, 2006.
9. J. H. Park, Robust stability of bidirectional associative memory neural networks with time delays, *Physics Letters A*, vol. 349, pp. 494-499, 2006
10. Zuo, Z.Q., Wang, Y.J.: Robust Stability Criteria of Uncertain Fuzzy Systems with Time-varying Delays. 2005 IEEE International Conference on Systems, Man and Cybernetics. (2005) 1303-1307
11. S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear matrix inequalities in systems and control theory*, Philadelphia, SIAM, 1994.
12. J. K. Hale and S. M. V. Lunel, *Introduction to functional differential equations*, Springer-Verlag, New York, 1993.
13. Zuo, Z.Q., Wang, Y.J.: Relaxed LMI condition for output feedback guaranteed cost control of uncertain discrete-time systems. *Journal of Optimization Theory and Applications*. **127** (2005) 207-217

# A Study into the Improvement of Binary Hopfield Networks for Map Coloring

Gloria Galán-Marín<sup>1</sup>, Enrique Mérida-Casermeiro<sup>2</sup>,  
Domingo López-Rodríguez<sup>2</sup>, and Juan M. Ortiz-de-Lazcano-Lobato<sup>3</sup>

<sup>1</sup> Department of Electronics and Electromechanical Engineering,  
University of Extremadura, Badajoz, Spain

gloriagm@unex.es

<sup>2</sup> Department of Applied Mathematics,  
University of Málaga, Málaga, Spain

{merida,dlopez}@ctima.uma.es

<sup>3</sup> Department of Computer Science and Artificial Intelligence,  
University of Málaga, Málaga, Spain

jmortiz@lcc.uma.es

**Abstract.** The map-coloring problem is a well known combinatorial optimization problem which frequently appears in mathematics, graph theory and artificial intelligence. This paper presents a study into the performance of some binary Hopfield networks with discrete dynamics for this classic problem. A number of instances have been simulated to demonstrate that only the proposed binary model provides optimal solutions. In addition, for large-scale maps an algorithm is presented to improve the local minima of the network by solving gradually growing submaps of the considered map. Simulation results for several  $n$ -region 4-color maps showed that the proposed neural algorithm converged to a correct colouring from at least 90% of initial states without the fine-tuning of parameters required in another Hopfield models.

## 1 Introduction

The  $k$ -coloring problem is a classic NP-complete optimization problem. The four color theorem states that any map drawn on a plane or sphere can be colored with four colors so that no two areas which share a border have the same color. The proof of this conjecture took more than one hundred years [1]. In 1976, Appel and Haken provided a computer-aided proof of the four-color theorem [2].

A powerful neural network for solving the map-coloring problem was presented by Takefuji and Lee [3]. The capability of the neural algorithm was demonstrated by solving examples of Appel and Haken's experiments through a large number of simulation runs. Remarkable solutions for many other combinatorial optimization problems have been presented by applying Takefuji and Lee's model, showing that it performs better than the best known algorithms [4,5,6]. Takefuji and Lee found discrete neurons computationally more efficient than continuous neurons [3]. Hence, they usually apply the continuous dynamics of the analog

Hopfield model with binary neurons. However, it has been recently demonstrated that this model does not always guarantee the descent of the energy function and can lead to inaccurate results and oscillatory behaviors in the convergence process [7,8,9,10].

In contrast, recently we have presented two binary Hopfield networks with discrete dynamics that always guarantee and maximize the decrease of the energy function. In the first one [11], a new input-output function is introduced into the binary Hopfield model with an asynchronous activation dynamics. Simulation results show that this sequential network converges to global optimal solutions for the n-queens problem [11]. However, since the operation of this model is based on the notion of single update, the required number of iteration steps for convergence is increased in proportion to the size of the problem. It led us to design a new binary neural network, the optimal competitive Hopfield model (OCHOM), based on the notion of group update [7]. It has been observed that the computation time is decreased even 100 times for large-scale networks comparing to the sequential model presented in [11]. In addition, performance comparison through massive simulation runs showed that for some problems the OCHOM is much superior to Takefuji and Lee's model in terms of both the solution quality and the computation time [7,8].

Recently, Wang and Tang [12] have improved the OCHOM by incorporating stochastic hill-climbing dynamics into the network. Simulation runs show that for some problems this algorithm obtains better solutions than the OCHOM, though the computation time is increased.

In this paper we study the performance of the binary neural networks through the four-color map problem. Despite the remarkable solutions obtained for some combinatorial optimization problems, simulation results show that the OCHOM network does not provide a global minimum solution for the map-coloring problem. Note that Joya et al. [13] proved that the Hopfield model with discrete dynamics never reached a correct solution for the k-colorability problem, while continuous dynamics succeeded to obtain a correct colouring. However, as pointed in [13], one major problem with the continuous model is that there is no analytical method to obtain the parameters of the network.

The major advantage of the OCHOM is that the search space is greatly reduced without a burden on the parameter tuning. However, it becomes a disadvantage for the map-coloring problem, where reducing the magnitude of the search space can easily bring on the problem of the local minimum convergence. It leads us to apply the sequential binary Hopfield model where more states of the network are allowed since every neuron can be activated on every step.

We have applied the binary sequential Hopfield model with the discrete input-output functions proposed by other authors [15,16]. It is confirmed by computer simulations that these networks never succeed to obtain a correct colouring. However, applying our discrete function [11] the sequential Hopfield network is capable of generating exact solutions for the four-color problem. On the other hand, simulation runs for large-scale maps show that the percentage of random initial states that do not converge to a global minimum is considerably increased.

Even for Hopfield networks with continuous dynamics it has been reported that with extremely large maps it is necessary to dynamically adjust the parameters so as to avoid local minima [14], where no method is given for this task.

We have modified the binary sequential network applying a hill-climbing term. However, we have found that this technique do not guarantee global minimum convergence for large-scale maps if the initial state is not a “correct” one. Hence, we have developed a method to avoid this difficulty without the fine-tuning of parameters required in another Hopfield models. The proposed algorithm solves a growing submap with the sequential network and uses as the initial state this optimal solution instead of a random binary state. Simulation runs in up to the 430-country map taken from the example of Appel and Haken’s experiment illustrate the effectivity and practicality of this neural approach.

## 2 Network Architecture

Let  $H$  be a binary neural network (1/0) with  $N$  neurons, where each neuron is connected to all the other neurons. The state of neuron  $i$  is denoted by  $v_i$  and its bias by  $\theta_i$ , for  $i = 1, \dots, N$ ;  $\omega_{ij}$  is a real number that represents the interconnection strength between neurons  $i$  and  $j$ , for  $i, j = 1, \dots, N$ . Note that we do not assume that self-connections  $\omega_{ii} = 0$ , as in the traditional discrete Hopfield model. Considering discrete-time dynamics, the Liapunov function of the neural network is given by:

$$E(k) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \omega_{ij} v_i(k) v_j(k) + \sum_{i=1}^N \theta_i v_i(k) \quad (1)$$

where  $k$  denotes discrete time. The inputs of the neurons are computed by the Hopfield’s updating rule:  $u_i(k) = \sum_{j=1}^N \omega_{ij} v_j(k) - \theta_i$ . We assume now that the network is updated asynchronously, that is, only one neuron  $i$  is selected for updating at time  $k$ . Hence, we have a sequential model where  $\Delta v_i \neq 0$ ,  $\Delta v_j = 0$ ,  $j = 1 \dots n$ ,  $j \neq i$ , and the energy change is:

$$\Delta E(k) = E(k+1) - E(k) = -\Delta v_i(k) \left[ u_i(k) + \frac{\omega_{ii}}{2} \Delta v_i(k) \right] \quad (2)$$

Since we have binary outputs  $v_i \in \{0, 1\}$ , it follows from (2) that:

- For  $\Delta v_i = 1$ ,  $\Delta E(k) \leq 0$  if and only if  $u_i(k) \geq -\frac{\omega_{ii}}{2}$
- For  $\Delta v_i = -1$ ,  $\Delta E(k) \leq 0$  if and only if  $u_i(k) \leq \frac{\omega_{ii}}{2}$

From these conditions we get that the energy is guaranteed to decrease if and only if the input-output function is:

$$v_i(k+1) = \begin{cases} 1 & \text{if } v_i(k) = 0 \text{ and } u_i(k) \geq -\frac{\omega_{ii}}{2} \\ 0 & \text{if } v_i(k) = 0 \text{ and } u_i(k) < -\frac{\omega_{ii}}{2} \\ 0 & \text{if } v_i(k) = 1 \text{ and } u_i(k) \leq \frac{\omega_{ii}}{2} \\ 1 & \text{if } v_i(k) = 1 \text{ and } u_i(k) > \frac{\omega_{ii}}{2} \end{cases} \quad (3)$$

For  $\omega_{ii} \geq 0$  the above expression reduces to:

$$v_i(k+1) = \begin{cases} 1 & \text{if } u_i(k) > \left| \frac{\omega_{ii}}{2} \right| \\ 0 & \text{if } u_i(k) < -\left| \frac{\omega_{ii}}{2} \right| \\ \text{change} & \text{if } -\left| \frac{\omega_{ii}}{2} \right| \leq u_i(k) \leq \left| \frac{\omega_{ii}}{2} \right| \end{cases}$$

and for  $\omega_{ii} < 0$  becomes to:

$$v_i(k+1) = \begin{cases} 1 & \text{if } u_i(k) \geq \left| \frac{\omega_{ii}}{2} \right| \\ 0 & \text{if } u_i(k) \leq -\left| \frac{\omega_{ii}}{2} \right| \\ \text{no change} & \text{if } -\left| \frac{\omega_{ii}}{2} \right| < u_i(k) < \left| \frac{\omega_{ii}}{2} \right| \end{cases} \quad (4)$$

Sun [15] proposed a generalized updating rule (GUR) for the binary Hopfield model updated in any sequence of updating modes. In the case of sequential mode this generalized updating rule is equivalent to the function proposed by Peng et al. in [16] and very similar to (4). Since these functions are only valid when  $\omega_{ii} < 0$ , it shows that (3) is not an instance of them.

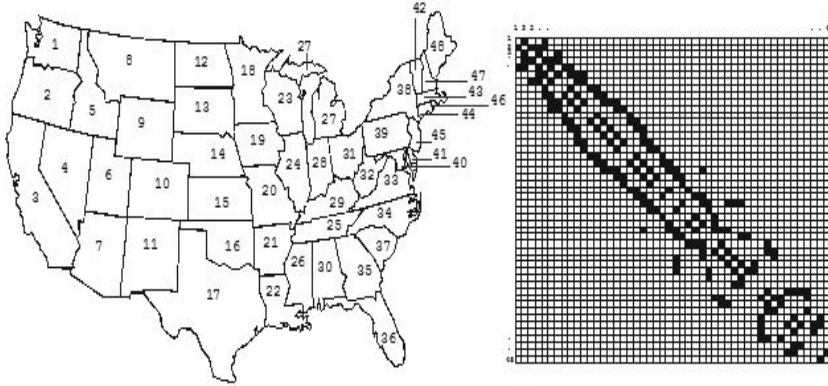
Observe that the function presented in [15][16] only differs from (4) in the case that  $u_i(k) = \left| \frac{\omega_{ii}}{2} \right|$  and  $v_i(k) = 0$  and in the case of  $u_i(k) = -\left| \frac{\omega_{ii}}{2} \right|$  and  $v_i(k) = 1$ . In these cases we have  $\Delta E = 0$  if we change the state of the neuron. Simulation runs in the four-coloring problem, a problem with  $\omega_{ii} < 0$ , show that only if we allow the network to evolve to another states with the same energy it is possible to reach the global minimum. For this reason our function (4) enables the network to generate a correct colouring. However, applying the function proposed in [15][16] the network is always trapped in unacceptable local minima.

### 3 The Proposed Algorithm for the Map-Coloring Problem

The neural network is composed of  $N \times K$  binary neurons, where  $N$  is the number of areas to be colored and  $K$  is the number of colors available for use in coloring the map. The binary output of the  $ikth$  neuron  $v_{ik} = 1$  means that color  $k$  is assigned to area  $i$ , and  $v_{ik} = 0$  otherwise. Hence, the energy function is defined:

$$E = B \sum_{i=1}^N \left( \sum_{k=1}^K v_{ik} - 1 \right)^2 + \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{k=1}^K a_{ij} v_{ik} v_{jk} \quad (5)$$

where  $B > 0$  is a constant which specifies the relative weighting of the first term and  $A = [a_{ij}]$  is the adjacency matrix which gives the boundary information between areas, that is,  $a_{ij} = 1$  if areas  $i$  and  $j$  have a common boundary, and  $a_{ij} = 0$  otherwise (see fig. 1). The first term in the energy function (5) becomes zero if one and only one neuron in each row has 1 as the output, and so a unique color is assigned to each area of the map. The second term vanishes when all neighboring areas do not have the same color. By comparing the energy function



**Fig. 1.** The U.S. continental map and its adjacency matrix given by a  $48 \times 48$  array. Black and white squares represent 1 and 0 values, respectively.

defined (5) and the Hopfield energy function (1), the connections weights and the biases are derived. If we substitute them in the Hopfield’s updating rule then:

$$u_{ik} = -\theta_{ik} + \sum_{j=1}^N \sum_{s=1}^K \omega_{ik,js} v_{js} = 2B - 2B \sum_{s=1}^K v_{is} - 2 \sum_{\substack{j=1 \\ j \neq i}}^N a_{ij} v_{jk} \quad (6)$$

Observe that the neural self-connection is  $\omega_{ik,ik} = -2B$  for all the neurons and then, according to (2), the energy change on every step is:

$$\Delta E = -u_{ik} \Delta v_{ik} + B(\Delta v_{ik})^2 \quad (7)$$

Simulation runs show that for medium-sized maps the proposed binary sequential model obtains a correct colouring from randomly generated initial states. However, for large-scale maps it is usually necessary a mechanism for improving local minima. The proposed neural algorithm is based upon the observed fact that the network can find more easily an exact solution for a given map using as the initial state an optimal solution of a submap of it, rather than using as the initial state a random binary matrix  $V$ .

In our algorithm on step  $n$  the neural network colors a map with  $n$  regions, using  $n \times K$  binary neurons and considering the adjacency matrix  $n \times n$ . On step  $n + 1$ , one region is added to the map and the network solves the  $(n + 1)$ -map using  $(n + 1) \times K$  binary neurons and considering the correct adjacency matrix  $(n + 1) \times (n + 1)$ . The algorithm introduces the optimal solution for the  $n$ -map problem obtained on step  $n$  as the initial state for the  $(n + 1)$ -map problem by completing with a  $1 \times K$  random binary vector the row  $n + 1$ . This does not mean that the optimal solution that we obtain on step  $n + 1$  must always include the solution obtained on step  $n$ , since the network can evolve to a different state.

Note also that the size of the network is modified with time and automatically adjusted to the size of the considered map.

Observe that in the algorithm  $NI_{max}$  denotes the maximum number of iteration steps for the time out procedure for solving the  $n$ -map. It means that if the network is trapped in a local minimum for the map with  $n$  regions, the algorithm automatically forces the network to solve the map with  $n + 1$  regions, and so on. For the 4-color problem, the network with  $n \times 4$  neurons is extended to a network with  $(n + 1) \times 4$  neurons, and so there is an increment of the energy function. Then, since the energy and the number of neurons are increased, there are more possible states with less or equal energy for the network to evolve in order to escape from local minima. The following procedure describes the algorithm proposed for solving the  $N$ -map  $K$ -coloring problem, in which we start with  $n = 1$  or  $n = \alpha \leq N$ , where  $\alpha$  is the number of regions in a solved submap:

1. Initialize the values of all the neuron outputs  $v_{ik}$ , for  $i = 1$  to  $n$  and  $k = 1$  to  $K$ , by randomly choosing 0 or 1.
2. Solve the  $n$ -map  $K$ -coloring problem:
  - (a) Extract from  $A$  ( $N \times N$ ) the adjacency matrix  $A_n(n \times n)$  for the  $n$ -map problem and set the number of iteration steps  $NI = 0$ .
  - (b) Evaluate the initial value of the energy function (5).
  - (c) Select randomly a neuron  $ik$ .
  - (d) Compute  $u_{ik}$ , the input of neuron  $ik$ , by eq. (6).
  - (e) Update the neuron output  $v_{ik}$  by the input-output function (4).
  - (f) Compute the energy change by eq. (7) and the new value of  $E$ .
  - (g) Increment the number of iteration steps by  $NI = NI + 1$ .
  - (h) Repeat from step 2.c until  $E = 0$  or  $NI = NI_{max}$ .
3. If  $n = N$  then terminate this procedure, else go to step 4.
4. Add a  $(1 \times K)$  random binary row vector to the optimal matrix  $V = [v_{ik}]_{n \times K}$  obtained in 2.
5. Increment  $n$  by  $n = n + 1$  and go to 2.

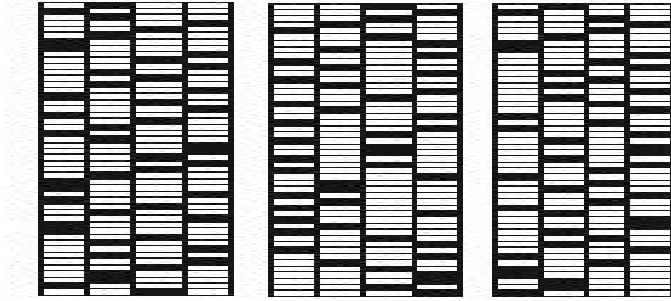
Observe that, if the network is still trapped in a local minimum when we reach  $n = N$ , that is, the full number of regions of the map, we can add imaginary regions to the real map until the network reaches the global minimum.

## 4 Simulation Results

We have tested the different binary Hopfield networks on the  $n$ -region 4-color maps solved by Takefuji and Lee in [3], that is, the U.S. continental map which consists of 48 states (see fig. 1) and the maps taken from the experiments of Appel and Haken [2]. Computational experiments were performed on an Origin 2000 Computer (Silicon Graphics Inc.) with 4 GBytes RAM by Matlab. For every map and network, we carried out 100 simulation runs from different randomly generated initial states. Simulation results showed that the OCHOM network [7,8] never reached a correct solution. Also, the binary model proposed in [15,16] was always trapped in unacceptable local minima for all the considered maps.



Initially, we have applied the proposed sequential model without the algorithm that solves gradually growing submaps. To prevent the network to assign no color to areas with a large number of neighbors, we strengthen the first term of the energy function by taking the constant value of the coefficient  $B = 2$ . Simulation runs for the U.S. map showed that the network converged to exact solutions, as the ones represented in fig. 2, from 95% of the random initial states. Then, we have found a few initial states from which the network is trapped in local minima. We have modified the network applying to our model a hill-climbing term. However, simulation runs in all the considered maps show that this technique does not either guarantee global minimum convergence.



**Fig. 2.** Three different solutions  $[V]_{48 \times 4}$  of the 48-state U.S. map 4-color problem provided by the sequential network. Black and white rectangles indicate 1 and 0 outputs, respectively, where the output  $v_{ik} = 1$  means that color  $k$  is assigned to state  $i$ .

We consider now a 210-country map taken from Appel and Haken's experiments and extract arbitrary submaps of 50, 70, 100 and 150 countries, where each one includes the one before. For the 50-map the network converged to an exact solution from 94 % of the random initial states, and for the 70-map from 92% of the random initial states. When we consider the maps with 100, 150 and 210 countries it is confirmed that for these similarly difficult maps the percentage of random initial states that converges to a correct colouring decreases with the size of the map. Therefore, for the 430-country map taken from Appel and Haken's experiments it is a rather difficult task finding randomly an initial state from which the network reaches a global minimum. Hence, it becomes necessary for extremely large maps to complete the network with the proposed algorithm to find adequate initial states from submaps.

When we applied the complete algorithm described in Sect. 3 with  $n = 1$ , the sequential model converged to an exact solution for the U.S. map from 100% of the initial states for a total of 100 runs. Also, for the 210-map of Appel and Haken the percentage of network simulations that produced a correct coloring was 100%. Observe that the network gradually colors all the extracted submaps, in this case 209 maps. Finally, when we apply this algorithm to the 430-map of Appel and Haken, 90 % of the network simulations provided a correct coloring

for a total of 100 runs. This percentage can be increased if we add imaginary regions to the real map as described in Sect. 3.

## 5 Conclusions

A study into the performance of different binary Hopfield networks through the four-coloring problem has been presented. Simulation results show that both the optimal competitive Hopfield model [7,8] and the sequential Hopfield model presented in [15,16] never reached a correct colouring. However, the proposed binary sequential Hopfield model is capable of generating exact solutions for medium-sized maps. Nevertheless, simulation results also show that the percentage of random initial states that converges to a correct colouring decreases with the size of the map when we consider similarly difficult maps. Hence, a neural algorithm is presented for large-scale maps to help the sequential model to find adequate initial states by solving submaps of the considered map. A number of instances have been simulated showing that this network provides an efficient and practical approach to solve the four-coloring problem even for large-scale maps without a burden on the parameter tuning.

## References

1. Saaty, T., Hainen, P.: The four color theorem: Assault and Conquest. Mc Graw-Hill (1977).
2. Appel, K., Haken, W.: The solution of the four-color-map problem, Scientific American, Oct. (1977) 108-121.
3. Takefuji, Y., Lee, K. C.: Artificial neural networks for four-colouring map problems and K-colorability problems. IEEE Trans. Circuits Syst. **38** (1991) 326-333.
4. Funabiki, N., Takenaka, Y., Nishikawa, S.: A maximum neural network approach for N-queens problem. Biol. Cybern. **76** (1997) 251-255.
5. Funabiki, N., Takefuji, Y., Lee, K. C.: A Neural Network Model for Finding a Near-Maximum Clique. J. of Parallel and Distributed Computing **14** (1992) 340-344.
6. Lee, K. C., Funabiki, N., Takefuji, Y.: A Parallel Improvement Algorithm for the Bipartite Subgraph Problem. IEEE Trans. Neural Networks **3** (1992) 139-145.
7. Galán-Marín, G., Muñoz-Pérez, J.: Design and Analysis of Maximum Hopfield Networks. IEEE Transactions on Neural Networks **12** (2001) 329-339.
8. Galán-Marín, G., Mérida-Casermeyro, E., Muñoz-Pérez, J.: Modelling competitive Hopfield networks for the maximum clique problem. Computers & Operations Research **30** (2003) 603-624.
9. Wang, L.: Discrete-time convergence theory and updating rules for neural networks with energy functions. IEEE Trans. Neural Networks **8** (1997) pp. 445-447.
10. Tateishi, M., Tamura, S.: Comments on 'Artificial neural networks for four-colouring map problems and K-colorability problems'. IEEE Trans. Circuits Syst. I: Fundamental Theory Applcat. **41** (1994) 248-249.
11. Galán-Marín, G., Muñoz-Pérez, J.: A new input-output function for binary Hopfield Neural Networks. Lecture Notes in Computer Science, Vol. 1606 (1999) 311-20.
12. Wang, J., Tang, Z.: An improved optimal competitive Hopfield network for bipartite subgraph problems. Neurocomputing **61** (2004) 413-419.

13. Joya, G., Atencia, M. A., Sandoval, F.: Hopfield neural networks for optimization: study of the different dynamics. *Neurocomputing* **43** (2002) 219-237.
14. Dahl, E. D.: Neural Network algorithm for an NP-Complete problem: Map and graph coloring. *Proc. First Int. Joint Conf. on Neural Networks* **III** (1987) 113-120.
15. Sun, Y.: A generalized updating rule for modified Hopfield neural network for quadratic optimization. *Neurocomputing* **19** (1998) 133-143.
16. Peng, M., Gupta, N. K., Armitage, A. F.: An investigation into the improvement of local minima of the Hopfield network. *Neural Networks* **9** (1996) 1241-1253.

# Automatic Diagnosis of the Footprint Pathologies Based on Neural Networks

Marco Mora<sup>1</sup>, Mary Carmen Jarur<sup>1</sup>, and Daniel Sbarbaro<sup>2</sup>

<sup>1</sup> Department of Computer Science, Catholic University of Maule  
Casilla 617, Talca, Chile

mora@spock.ucm.cl, mjarur@spock.ucm.cl

<http://www.ganimides.ucm.cl/mmora/>

<sup>2</sup> Department of Electrical Engineering, University of Concepcion  
Casilla 160-C, Concepcion, Chile

dsbarbar@die.udec.cl

**Abstract.** Currently foot pathologies, like cave and flat foot, are detected by a human expert who interprets a footprint image. The lack of trained personal to carry out massive first screening detection campaigns precludes the routinary diagnostic of these pathologies. This work presents a novel automatic system, based on Neural Networks (NN), for foot pathologies detection. In order to improve the efficiency of the neural network training algorithm, we propose the use of principal components analysis to reduce the number of inputs to the NN. The results obtained with this system demonstrate the feasibility of building automatic diagnosis systems based on the foot image. These systems are very valuable in remote areas and can be also used for massive first screening health campaigns.

## 1 Introduction

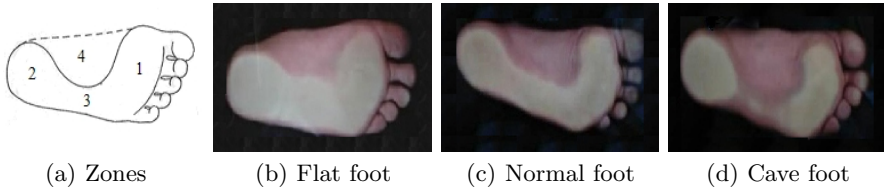
The cave foot and the flat foot are pathologies presented in children once they are three year old. If these foot malformations are not detected and treated on time, they get worst during adulthood producing several disturbance, pain and posture-related disorders.

When the foot is planted, not all the sole is in contact with the ground. The footprint is the surface of the foot in contact with the ground. The characteristic form and zones of the footprint are shown in Fig. 1a. Zones 1, 2 and 3 correspond to regions in contact with the surface when the foot is planted, these are called anterior heel, posterior heel and isthmus respectively. Zone 4 does not form part of the surface in contact and is called footprint vault [7].

In the diagnosis of foot pathologies, considering the footprint form, it is possible to use computational methods based on digital image processing, in order to classify the different types of anomalies. In case of pathologies such as flat foot, cave foot and others, the computational detection of the footprint allows the development of automatic systems for diagnosis. These systems will substantially decrease the time between the test execution and the examination results. This

work describes the development of an automatic method for foot pathologies detection based on the use neural networks and principal components analysis.

It is possible to classify a foot by its footprint form and dimensions as a: normal, flat or cave foot. In this classification an important role is played by the ratio between the distance of the isthmus's thinnest zone and the distance of the anterior heel's widest zone. Considering the values of this ratio, the foot can be classified as flat foot (Fig. 1b), normal (Fig. 1c) foot and cave foot (Fig. 1d).



**Fig. 1.** Images of the sole

Currently, a human expert defines if a patient has a normal, cave or flat foot by a manual exam called photopodogram. A photopodogram is a chemical photo of the foot part supporting the load. The expert determines the position for the two distances, sizes them, calculates the ratio and classifies the foot. Even though the criteria for classifying footprints seems very simple, the use of a classifier based on neural networks (NN) offers the following advantages compared with more traditional approaches: (1) it is not simple to develop an algorithm to determine with precision the right position to measure the distances, and (2) it can be trained to recognize other pathologies or to improve their performance as more cases are available.

The multilayer perceptron (MLP) and the training algorithm called back-propagation (BP) [6] have been successfully used in classification and functional approximation. An important characteristic of MLP is its capacity to classify patterns grouped in classes not lineally separable. Besides that, there are powerful tools, such as the Levenberg-Marquardt optimization algorithm [1], and a Bayesian approach for defining the regularization parameters [3], which enable the efficient training of MLP. Even though there exist this universal framework for building classifiers, as we will illustrate in this work, a simple preprocessing can lead to smaller network structures without compromising performance.

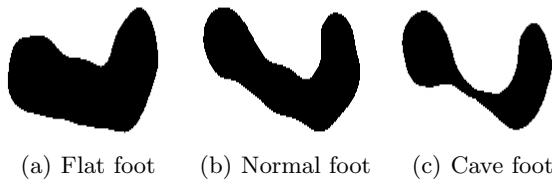
An instrument, consisting of a robust metallic structure with adjustable height and transparent glass in its upper part, was developed for acquiring the footprints images. The patient must stand on the glass and the footprint image is obtained with a digital color camera in the interior of the structure. For adequate lighting, white bulbs are used. The system includes the regulation of the light intensity of the bulbs, which allows the amount of light to be adjusted for capturing

images under different lighting conditions. This system has been used to build a database of color images, consisting of more than 230 images of children feet age between 9 and 12. These images were classified by an expert<sup>1</sup>. From the total sample, 12.7% are flat feet, 61.6% are normal feet and 25.7% are cave feet.

Matlab and the Neural Networks Toolbox were used as platform for carrying out most of data processing work. This paper is organized as follows. Section 2 describes the footprint representation and characteristics extraction. Section 3 presents the training of the neural network classifier. Section 4 shows the validation of the neural network classifier. Finally, Sect. 5 shows some conclusions and future studies.

## 2 Footprint Representation and Characteristics Extraction

Prior to classification, the footprint is isolated from the rest of the components of the sole image by using the method proposed in [4]. Figures 2a, 2b and 2c shown the segmentation of a flat, a normal foot, and a cave foot respectively.

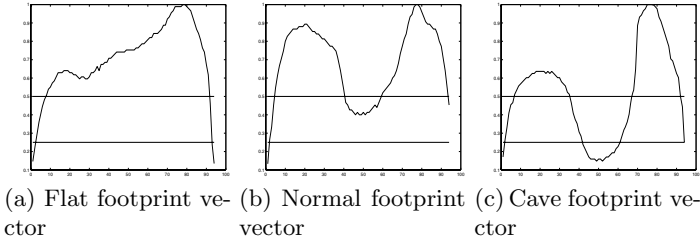


**Fig. 2.** Segmentation of footprint without toes

After performing the segmentation, the footprint is represented by a vector containing the width in pixels of the segmented footprint, without toes, by each column in the horizontal direction. Because every image has a width vector with different length, the vectors were normalized to have the same length. Also the value of each element was normalized to a value in the range of 0 to 1. Figures 3a, 3b and 3c show the normalized vectors of a flat, a normal and a cave foot.

As a method to reduce the dimensionality of the inputs to the classifier, a principal components analysis was used [2]. Given an eigenvalue  $\lambda_i$  associated to the covariance matrix of the width vector set, the percentage contribution  $\gamma_i$  [1] and the accumulated percentage contribution  $APC_i$  [2] are calculated by the following expressions:

<sup>1</sup> The authors of this study acknowledge Mr. Eduardo ACHU, specialist in Kinesiology, Department of Kinesiology, Catholic University of Maule, Talca, Chile, for his participation as an expert in the classification of the database images.



**Fig. 3.** Representation of the footprint without toes

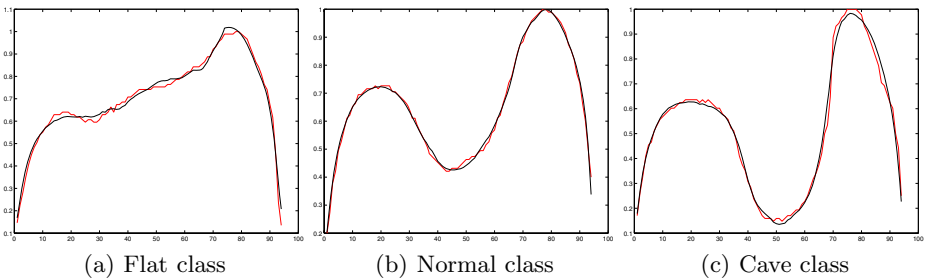
$$\gamma_i = \frac{\lambda_i}{\sum_{j=1}^d \lambda_j} \tag{1}$$

$$APC_i = \sum_{j=1}^i \gamma_j . \tag{2}$$

Table 1 shows the value, percentage contribution and the accumulated percentage contribution of the first nine eigenvalues. It is possible to note that from the 8<sup>th</sup> eigenvalue the contribution is close to zero, and then it is enough to represent the width vector with the first seven principal components. Figure 4 shows a normalized width vector (rugged red signal) and the resultant approximation from using the seven first main components (smoothed black signal) for the three classes.

**Table 1.** Contribution of the first 9 eigenvalues

-	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\lambda_5$	$\lambda_6$	$\lambda_7$	$\lambda_8$	$\lambda_9$
Value	0.991	0.160	0.139	0.078	0.055	0.0352	0.020	0.014	0.010
Percentual contribution	63.44	10.25	8.95	5.01	3.57	2.25	1.31	0.94	0.65
Accumulated contribution	63.44	73.7	82.65	87.67	91.24	93.50	94.82	95.76	96.42



**Fig. 4.** Principal components approximation

### 3 Training of the Neural Network Classifier

A preliminary analysis of the segmented footprint analysis showed very little presence of limit patterns among classes: flat feet almost normal, normal feet almost flat, normal feet almost cave and cave feet almost normal. Thus, the training set was enhanced with 4 synthetic patterns for each one of the limit cases. Thus the training set has a total of 199 images, 12.5% corresponding to a flat foot, 63% to a normal one and 24.5% to a cave foot.

To build the training set the first seven principal components were calculated for all the width vectors in the training set. For the foot classification as a flat, normal or cave foot, a MLP trained with Bayesian regularization backpropagation was used. The structure of the NN is:

- Number of inputs: 7, one for each main component.
- Number of outputs: 1. It takes a value of 1 if the foot is flat, a value of 0 when the foot is normal and a value of  $-1$  when the foot is cave.

To determine the amount of neurons of the hidden layer, the procedure described in [1] was followed. Batch learning was adopted and the initial network weights were generated by the Nguyen-Widrow method [5] since it increases the convergence speed of the training algorithm.

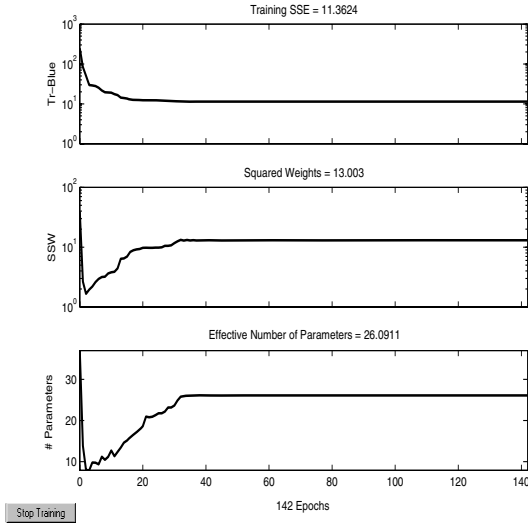
The details of this procedure are shown in Table 2, where NNCO corresponds to the number of neurons in the hidden layer, SSE is the sum squared error and SSW is the sum squared weights. From the Table 2 it can be seen that from 4 neurons in the hidden layer, the SSE, SSW and the effective parameters stay practically constants. As a result, 4 neurons are considered in the hidden layer.

**Table 2.** Determining the amount of neurons in the hidden layer

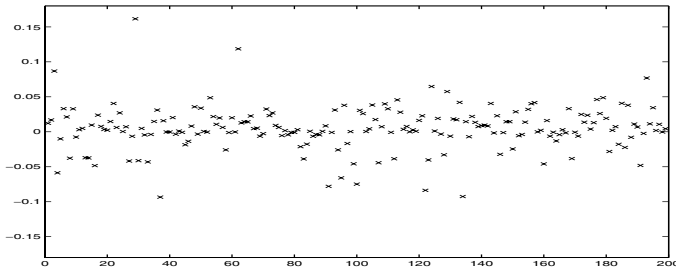
NNCO	Epochs	SSE	SSW	Effective parameters	Total parameters
1	114/1000	22.0396/0.001	23.38	8.49e+000	10
2	51/1000	12.4639/0.001	9.854	1.63e+001	19
3	83/1000	12.3316/0.001	9.661	1.97e+001	28
4	142/1000	11.3624/0.001	13.00	2.61e+001	37
5	406/1000	11.3263/0.001	13.39	2.87e+001	46
6	227/1000	11.3672/0.001	12.92	2.62e+001	55

In Fig. 5 it is possible to observe that the SSE, SSW and the effective parameters of the network are relatively constant over several iterations, this means that the training process has been appropriately made. Figure 6 shows the training error by each pattern of the training set. From the figure it is important to emphasize that the classification errors are not very small values. This behavior assures that the network has not memorized the training set, and it will generalize well.





**Fig. 5.** Evolution of the training process for 4 neurons in the hidden layer. Top: SSE evolution. Center: SSW evolution. Down: Effective parameters evolution.



**Fig. 6.** Classification error of the training set

## 4 Validation of the Neural Network Classifier

The validation set contains 38 new real footprint images classified by the expert, where 13.1% correspond to a flat foot, 55.3% to a normal one and 31.6% to a cave foot. For each footprint in the validation set, the corresponding normalized-width vector was calculated for the binary images of the segmented footprint, and then by performing principal component decomposition only the first 7 axes were presented to the trained NN. The Fig. 7 shows the results of the

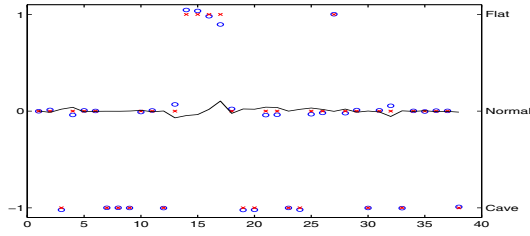


Fig. 7. Classification error of validation set and output/target of the net

classification, the outputs of the network and the targets are represented by circles and crosses respectively. Moreover the figure shows the error of the classification represented by a black continuous line. The results are very good, considering that the classification was correct for the complete set.

In order to illustrate the whole process, we have chosen 9 patterns of the validation set. Figure 8 shows the foot images and its segmented footprints. Figures 8a-c, 8d-f and 8g-i correspond to flat, normal and cave feet respectively. Figure 9 shows the footprint vectors of the previous images.

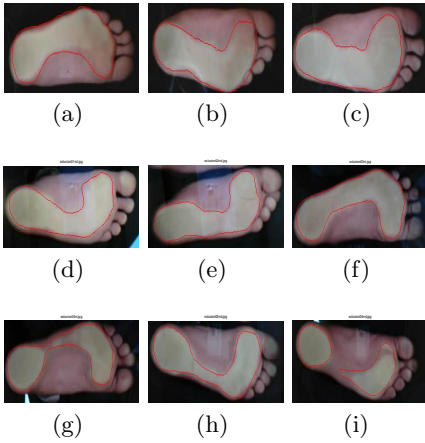


Fig. 8. Foot plants and its segmented footprints

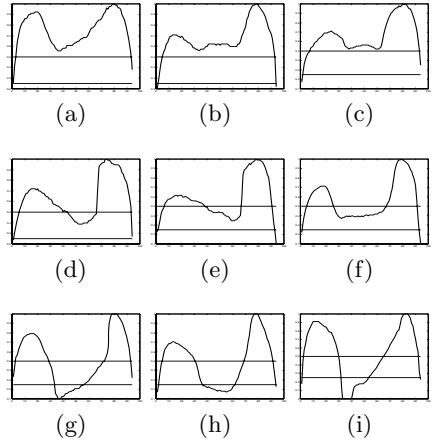


Fig. 9. Footprint vectors of the segmented footprint

Table 3 shows quantitative classification results of the 9 examples selected from the validation set, as can be seen the network classification and the expert classification are equivalent.

**Table 3.** Classification of validation set

ID footprint	Output Net	Target Output	Error of classification	Classification of Net	Classification of expert
(a)	1.0457	1	-0.0457	Flat	Flat
(b)	1.0355	1	-0.0355	Flat	Flat
(c)	0.8958	1	0.1042	Flat	Flat
(d)	0.0126	0	-0.0126	Normal	Normal
(e)	-0.0395	0	0.0395	Normal	Normal
(f)	0.0080	0	-0.0080	Normal	Normal
(g)	-0.9992	-1	-0.0008	Cave	Cave
(h)	-1.0010	-1	0.0010	Cave	Cave
(i)	-0.9991	-1	-0.0009	Cave	Cave

## 5 Final Remarks and Future Studies

This work has presented a method to detect footprint pathologies based on neural networks and principal components analysis. Our work shows a robust solution to a real world problem, in addition it contributes to automate a process that currently is made by a human expert.

By adding synthetic border patterns, the training process was enhanced. The footprint representation by a width vector and principal component analysis were used. By using a MLP trained by a Bayesian approach, all patterns of the validation set were correctly classified.

The encouraging results of this study demonstrate the feasibility of implementing a system for early, automatic and massive diagnosis of the pathologies analyzed in this study. In addition, this study lay down the foundation for incorporating new foot pathologies, which can be diagnosed from the footprint.

Finally, considering the experience obtained from this study, our interest is centered in the real time footprint segmentation for monitoring, analysis and detection of walking disorders.

## References

1. Foresee D. and Hagan M., "Gauss-Newton Approximation to Bayesian Learning", in Proceedings of the International Joint Conference on Neural Networks, 1997.
2. Jolliffe I., "Principal Component Analysis", Springer-Verlag, 1986.
3. Mackay D., "Bayesian Interpolation", Neural Computation, vol.4, no.3, 1992.
4. Mora M., Sbarbaro D., "A Robust Footprint Detection Using Color Images and Neural Networks", in Proceedings of the CIARP 2005, Lecture Notes in Computer Science, vol. 3773, pp. 311-318, 2005.
5. Nguyen D. and Widrow B., "Improving the Learning Speed of 2-Layer Neural Networks by Choosong Initial Values of the Adaptive Weights", in Proceedings of the IJCNN, vol. 3, pp. 21-26, 1990.
6. Rumelhart D., McClelland J. and PDP group, "Explorations in Parallel Distributed Processing", The MIT Press. vol. 1 and 2, 1986.
7. Valenti V., "Orthotic Treatment of Walk Alterations", Panamerican Medicine, (in spanish) 1979.

# Mining Data from a Metallurgical Process by a Novel Neural Network Pruning Method

Henrik Saxén<sup>1</sup>, Frank Pettersson<sup>1</sup>, and Matias Waller<sup>2</sup>

<sup>1</sup> Heat Engineering Lab., Åbo Akademi University, Biskopsg. 8, 20500 Åbo, Finland  
{Henrik.Saxen, Frank.Pettersson}@abo.fi

<sup>2</sup> Åland Polytechnic, PB 1010, AX-22111 Mariehamn, Åland (Finland)  
Matias.Waller@ha.ax

**Abstract.** Many metallurgical processes are complex and due to hostile environment it is difficult to carry out reliable measurement of their internal state, but the demands on high productivity and consideration of environmental issues require that the processes still be strictly controlled. Due to the complexity and non-ideality of the processes, it is often not feasible to develop mechanistic models. An alternative is to use neural networks as black-box models, built on historical process data. The selection of relevant inputs and appropriate network structure are still problematic issues. The present work addresses these two problems in the modeling of the hot metal silicon content in the blast furnace. An algorithm is applied to find relevant inputs and their time lags, as well as a proper network size, by pruning a large network. The resulting models exhibit good prediction capabilities and the inputs and time lags detected are in good agreement with practical metallurgical knowledge.

## 1 Introduction

Neural networks have become popular nonlinear modeling tools due to their universal approximation capabilities [1], but their use is not always straightforward. A typical problem is that there are often far more inputs, which potentially influence the output (dependent) variable, than can be considered in a final parsimonious model, so a choice between the inputs has to be made to avoid over-parameterization [2]. Another problem is that practical measurements always contain errors, so the number of network parameters has to be restricted to avoid fitting noise. Several algorithms (see, e.g., [3-5]) with either growing or pruned networks have been proposed but many of these are based on heuristics for noise-free cases, or include retraining steps that require prohibitive computational efforts. Some general methods, which do not take a stand on the required complexity of the model, have been proposed for selection of relevant inputs [6], and recently the authors of the present paper proposed an efficient pruning method based on neural networks [7] where both the relevant inputs and the network connectivity are detected. This algorithm is in the present paper applied to a complex time-series modeling problem from the metallurgical industry, where relevant inputs and time lags of these have to be detected in the modeling. The algorithm is outlined in Section 2, and applied in Section 3 to the hot metal silicon

prediction problem. It is found to yield parsimonious models in agreement with process knowledge. Section 4 finally presents some conclusions.

## 2 The Method

### 2.1 The Pruning Algorithm

The pruning algorithm is based on feedforward neural networks of multi-layer perceptron type with a single layer of hidden nonlinear units and a single linear output node. Practical experience has shown that such networks with an arbitrary choice of weights in their lower layer of connections,  $\mathbf{W}$ , can provide a relatively good solution of the approximation problem at hand if the upper-layer weight vector,  $\mathbf{v}$ , is chosen properly. The vector  $\mathbf{v}$ , in turn, can be determined simply by a matrix inversion. With this as the starting point, the pruning algorithm is outlined:

1. Select a set of  $N$  potential inputs,  $\mathbf{x}$ , and the output,  $y$ , to be estimated for the observations of the training set.
2. Choose a sufficient number of hidden nodes,  $m$ , and generate a random weight matrix,  $\mathbf{W}^{(0)}$ , for the lower part of the network. Set the iteration index to  $k = 1$ .
3. Equate to zero, in turn, each non-zero weight,  $w_{ij}^{(k-1)}$ , of  $\mathbf{W}^{(k-1)}$ , and determine the optimal upper-layer weight vector,  $\mathbf{V}$ , by linear least squares. Save the corresponding value of the objective function,  $F_{ij}^{(k)} = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ .
4. Find the minimum of the objective function values,  $F(k) = \min_{ij}\{F_{ij}^{(k)}\}$ . Set  $\mathbf{W}^{(k)} = \mathbf{W}^{(k-1)}$  and equate to zero the weight corresponding to the minimum objective function value,  $w_{\tilde{i}\tilde{j}}^{(k)} = 0$  with  $\tilde{i}\tilde{j} = \arg \min_{ij}\{F_{ij}^{(k)}\}$ .
5. Set  $\psi_{\tilde{i}\tilde{j}} = k$  and save this variable in a matrix,  $\mathbf{\Psi} = \{\psi_{\tilde{i}\tilde{j}}\}$  (with the same dimension as  $\mathbf{W}$ ).
6. Set  $k = k + 1$ . If  $k < m \cdot N$ , go to 3. Else, end.

$\mathbf{\Psi}$  here stores the iteration number at which each connection weight has been deleted, and by studying the elements in the columns (or rows) of the matrix one can deduce when a certain input (or hidden node) was eliminated.

The computational effort of the method can be reduced considerably by some simple measures. The order in which the weights are studied in the algorithm is unimportant, so one can go through them in the order of the hidden nodes they refer to (i.e., starting with the connections between the inputs and the first hidden node, etc.). First, the net input to each hidden node,  $ai(t) = \sum_{j=1}^N w_{ij}x_j(t)$ ,  $i=1, \dots, m$ , at each "time step",  $t$ , is determined, as well as the corresponding output,  $z_i = \sigma(a_i)$ . At step 3 of the algorithm, a resetting of  $w_{ik}$  simply means that the net input of the  $i^{\text{th}}$  hidden node is changed into  $a_i(t) - w_{ik}x_k(t)$ , while the net inputs, and outputs, of all other hidden nodes remain unaltered. Thus, for each weight in step 3, only one multiplication and one subtraction is needed to get the net input. In addition, the nonlinear (sigmoidal) transformation is required for the hidden node in question.

## 2.2 An Example

Consider a data set with a ten-dimensional input vector ( $N = 10$ ) of normally distributed random values with zero mean and unit variance and the dependent variable given by

$$y = x_2 - 3x_4^2 + 2x_5x_7 \quad (1)$$

Six of the ten inputs are useless variables for capturing the input-output relation at hand. Training and test sets of 250 observations each were generated, and the algorithm was run using a network with  $m = 10$  hidden nodes. Fig. 1 illustrates how the errors on the training set (solid line) and test set (dashed line) evolve during the pruning process, progressing from right to left. The errors are seen to decrease in a step-wise manner. The lower panel shows the final region in better focus.

Closer study of the model with a lower-layer complexity of eight connections (indicated by a circle in the lower panel of Fig. 1) reveals that it is a network with seven hidden nodes and only four inputs,  $x_2, x_4, x_5$  and  $x_7$ , i.e., only the relevant ones. It has a sparse and an intuitively appealing connectivity, where three inputs ( $x_2, x_5$  and  $x_7$ ) are connected to a hidden node each, a joint hidden node is used in the approximation of the product between  $x_5$  and  $x_7$ , and three hidden nodes are devoted to the approximation of the square of  $x_4$ . Such a sparse network lends itself perfectly to a deeper analysis of how it constructs its approximation of the nonlinearities [8].

## 2.3 Extension to On-Line Learning

A shortcoming of nonlinear black-box models is that they may yield poor predictions on independent data if some input exhibits drift or sudden level changes. A remedy follows logically from the over-all approach taken in the algorithm: It is natural to adjust the upper-layer weights as new information enters, and this is a linear problem, since a given input vector,  $\mathbf{x}(t)$ , and a fixed set of lower-layer weights yield a fixed output,  $\mathbf{z}(t)$ , from the hidden nodes for every time instant  $t$ . For updating the upper-layer weights,  $\mathbf{v}$ , the well known Kalman filter [9] is used

$$\mathbf{v}(t+1) = \mathbf{v}(t) + K(t+1)(y(t) - \hat{y}(t)) \quad (2)$$

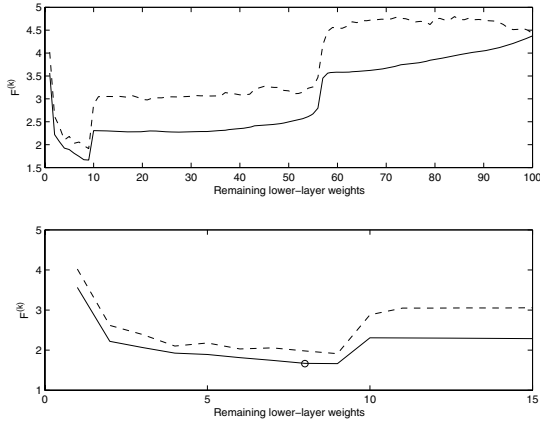
where the Kalman gain is given by

$$K(t+1) = \frac{\mathbf{P}(t)\mathbf{z}^T(t+1)}{1 + \mathbf{z}(t+1)\mathbf{P}(t)\mathbf{z}^T(t+1)} \quad (3)$$

while the matrix  $\mathbf{P}$  is updated by

$$\mathbf{P}(t+1) = \mathbf{P}(t) + \mathbf{R} - K(t+1)\mathbf{z}(t+1)\mathbf{P}(t) \quad (4)$$

Without a priori information, the covariance matrix of the “measurement error”,  $\mathbf{R}$ , is often chosen as a diagonal matrix,  $c\mathbf{I}$ , where a small  $c$  gives a high gain,  $K$ , which, in turn, makes large weight changes possible. The updating can be made less sensitive to the initial value of the matrix,  $\mathbf{P}(1)$ , by first applying the filter on the training set, using the final weights and matrices as starting points for the predictions.



**Fig. 1.** Training (—) and test (- -) errors for the example problem as functions of the number of remaining weights (excluding biases) in the lower part of the network

### 3 Application to Hot Metal Silicon Prediction

#### 3.1 The Prediction Problem and Process Data

The blast furnace is the principal unit in the most important process route for iron produced for primary steelmaking. It acts as a large counter current chemical reactor and heat exchanger [10]. At its top the main energy source, coke, is charged together with preprocessed ore and fluxes in alternating layers. The ore is heated, reduced and finally smelted by the ascending gases, and intermittently tapped out at the bottom of the furnace in the form of liquid iron (often called hot metal). Large time delays and sluggish response to control actions make it important to predict quality and operational variables, e.g., the composition of the hot metal. The silicon content of the hot metal is an important indicator of the thermal state of the furnace: A decreasing silicon content often indicates a cooling of the furnace that, without due countermeasures, can lead to serious operational complications, while a high silicon content indicates excessive generation of heat and waste of coke. Blast furnaces are usually operated with a safety margin, i.e., a slightly higher coke rate than is deemed necessary, but since the cost of coke is dominating in ironmaking, there are obvious economical benefits of making the safety margin smaller. This requires stricter control of the heat level, and numerous models for the prediction of the hot metal silicon content have therefore been developed [11-19]. In [16] an exhaustive search was made among linear FIR models using inputs with different time lags and [18] applied a partial least squares procedure, but in the papers on silicon prediction by neural networks a small set of potential inputs was always selected a priori.

The present analysis was based on a data set from a Swedish blast furnace. All variables were preprocessed to yield hourly mean values. The input dimension was limited to 15 on the basis of process knowledge: These inputs were the total and specific blast volume, pressure and temperature, the oxygen enrichment, the flame temperature, bosh permeability, the (calculated) specific coke and coal rates, the

energy available at the tuyeres, the cooling losses of the tuyeres, the solution loss coke, the top gas CO utilization, the sum of the carbon monoxide and dioxide content, and the ore-to-coke ratio. Specific quantities mentioned above are expressed per ton of hot metal.

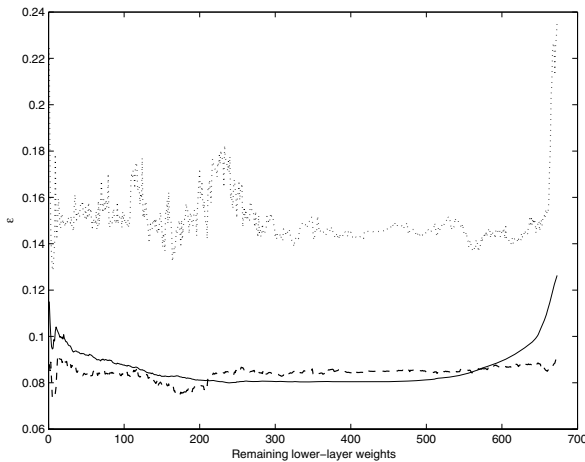
To evaluate the potential of on-line prediction, the problem was written as

$$\hat{y}(t) = f(\mathbf{x}(t), \mathbf{x}(t-1), \mathbf{x}(t-2), \dots, \mathbf{x}(t-8)) \tag{5}$$

including lags of the inputs,  $\mathbf{x}$ , up to eight hours. The dimension of the input vector is thus 135 (= 15 × 9). Autoregressive terms (i.e.  $y(t-1)$ ,  $y(t-2)$ ,.. on the RHS) were not considered since the inclusion of such is known to yield models of high inertia and with small possibilities to predict rapid changes [16]. Data for 800 h of operation was used for training, while the roughly 190-h period that followed was used for model evaluation, first normalizing all variables to (0,1).

### 3.2 Results

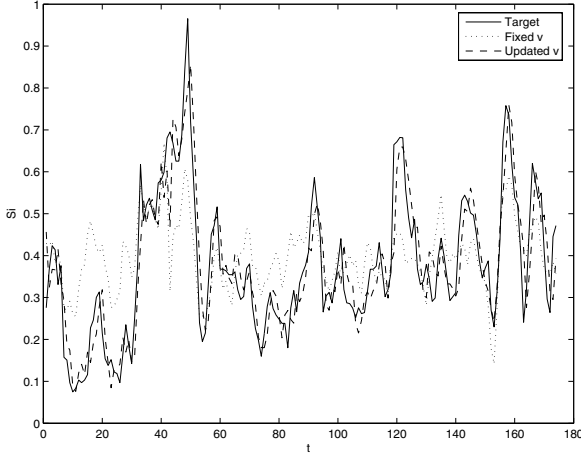
The performance of the algorithm is illustrated by a typical run of it from a random starting weight matrix, using a network with five hidden nodes. The Kalman filter was initialized by  $\mathbf{P}(1) = 10 \mathbf{I}$  (at the start of the training set) while  $\mathbf{R} = \mathbf{I}$ . Figure 2 illustrates the evolution of the root mean square errors,  $\epsilon = \sqrt{F/n}$ , where  $n$  is the number of observations, on the training set (solid lines), test set without (dotted lines) and with (thick dashed lines) online adaptation of the upper-layer weights. The algorithm is seen to initially reduce both the training and test errors (without online learning) considerably, but the latter remains on a considerably higher level throughout the pruning. By contrast, the continuously updated model only shows a slowly decreasing trend, but yields errors comparable with the training errors.



**Fig. 2.** Training errors (solid lines), test errors without (dotted lines) and with weight updating (dashed lines) as functions of the number of remaining weights in the lower part of the network for the silicon prediction problem



From the run it can be deduced that a complexity of about six connections in the lower layer would be required. Figure 3 illustrates the target of the test set (solid line), the approximation provided by this network with fixed upper layer weights (dotted lines) as well as the one given by the network with online learning (dashed lines): The latter approximation is seen to be clearly superior.



**Fig. 3.** (Normalized) silicon content (solid lines), prediction without (dotted lines) and with updating (dashed lines) of the weights for the test set

Taking this as the model candidate, it is interesting to study the inputs that were selected by the algorithm. The model can be written as

$$\hat{Si}(t) = f(Q_{\text{tuy}}(t-2), m_{\text{coal}}(t-1), m'_{\text{coal}}(t-1), \kappa(t-3), V'_{\text{bl}}(t-2)) \tag{6}$$

where  $f$  is a neural network with only two hidden nodes, the prime on a symbol denotes a specific quantity and the inputs have been listed in the order of importance. In this model, the heat loss at the tuyeres (lagged by two hours), the specific coal injection rate (lagged by one hour), and the gas permeability factor (lagged by three hours) are connected to the first hidden node, while the specific blast volume (lagged by two hours) and the absolute specific coal rate (lagged by one hour) are connected to the second hidden node. These input variables are related to the tuyere parameters, which are known to affect the silicon transfer in the furnace. The selected variables, as well as their comparatively short time lags (maximum three hours), are findings that agree well with knowledge from the practical operation of the furnace. In particular, the tuyere heat loss is known to reflect the intensity of the thermal conditions in the lower furnace, and the coal injection is a primary means of short-term control of the heat level of the furnace. The gas permeability, in turn, reflects the vertical extent of the high-temperature region.

In order to study more general features of the problem, a set of runs of the algorithm were undertaken: Starting from ten random initial weight matrices,  $\mathbf{W}^{(0)}$ , some scattering of the important inputs detected was, as expected, observed, but

many of the variables detected in the single run described above were present. A striking feature was that the last input variable to be eliminated was always  $Q_{\text{tuy}}$ , which shows the strong correlation between tuyere heat loss and the hot metal silicon content.

## 4 Conclusions

An algorithm for the selection of input variables, their time lags as well as a proper complexity of a multi-layer feedforward neural network has been developed based on an efficient pruning approach. A method for on-line updating of the upper-layer weights was also proposed. Applied to a problem in the metallurgical industry, i.e., the prediction of the silicon content of hot metal produced in a blast furnace, the method has been demonstrated to find inputs that are known to correlate with silicon content, and also to detect time lags that can be explained considering the dynamics of the process. The results hold promise for a practical application of the model in the automation system of the iron works.

## References

1. Cybenko, G.: Approximations by superpositions of sigmoidal function. *Math. Contr., Sign.* 2 (1989) 303-314.
2. Principe, J.C., Euliano, N.R., Lefebvre, W.C.: *Neural and adaptive systems: Fundamentals through simulations.* John Wiley & Sons, New York, (1999).
3. Frean, M.: The Upstart Algorithm. A Method for Constructing and Training Feed-forward Neural Networks. *Neural Computation* 2 (1991) 198-209.
4. Fahlman, S.E., Lebiere, C.: The Cascade-Correlation Learning Architecture. In: Touretzky, D.S. (ed.): *Adv. Neural Inf. Proc. Syst. 2.* Morgan Kaufmann (1990) 524-532.
5. Y. Le Chun, Y., Denker, J. S., Solla, S.A.: Optimal Brain Damage. In: Touretzky, D.S. (ed.): *Adv. Neural Inf. Proc. Syst. 2.* Morgan Kaufmann (1990) 598-605.
6. Sridhar, D.V., Bartlett, E.B., Seagrave, R.C.: Information theoretic subset selection for neural networks. *Comput. Chem. Engng.* 22 (1998) 613-626.
7. Saxén, H., Pettersson, F.: Method for the selection of inputs and structure of feedforward neural networks. *Comput. Chem. Engng.* 30 (2006) 1038-1045.
8. Hinnelä, J., Saxén, H., Pettersson, F.: Modeling of the blast furnace burden distribution by evolving neural networks. *Ind. Engng Chem. Res.* 42 (2003) 2314-2323.
9. Haykin, S.: *Kalman filtering and neural networks.* Wiley, New York (2001).
10. Omori, Y. (ed.): *Blast Furnace Phenomena and Modelling.* The Iron and Steel Institute of Japan, Elsevier, London, (1987).
11. Phadke, M.S., Wu, S.M.: Identification of Multiinput - Multioutput Transfer Function and Noise Model of a Blast Furnace from Closed-Loop Data. *IEEE Trans. Aut. Contr.* 19 (1974) 944-951.
12. Unbehauen, H., Diekmann, K.: Application of MIMO Identification to a Blast Furnace. *IFAC Identification and System Parameter Estimation* (1982) 180-185.
13. Saxén, H.: Short Term Prediction of Silicon Content in Pig Iron. *Can. Met. Quart.* 33 (1994) 319-326.
14. Saxén, H., Östermark, R.: State Realization with Exogenous Variables - A Test on Blast Furnace Data. *Europ. J. Oper. Res.* 89 (1996) 34-52.

15. Chen, J.: A Predictive System for Blast Furnaces by Integrating a Neural Network with Qualitative Analysis. *Engng. Appl. AI* 14 (2001) 77-85.
16. Waller, M., Saxén, H.: On the Development of Predictive Models with Applications to a Metallurgical Process. *Ind. Eng. Chem. Res.* 39 (2000) 982-988.
17. Waller, M., Saxén, H.: Application of Nonlinear Time Series Analysis to the Prediction of Silicon Content of Pig Iron. *ISIJ Int.* 42 (2002) 316-318.
18. Bhattacharya, T.: Prediction of silicon content in blast furnace hot metal using Partial Least Squares (PLS). *ISIJ Int.* 45 (2005) 1943-1945.
19. Gao, C.H., Qian, J.X.: Time-dependent fractal characteristics on time series of silicon content in hot metal of blast furnace. *ISIJ Int.* 45 (2005) 1269-1271.

# Dynamic Ridge Polynomial Neural Networks in Exchange Rates Time Series Forecasting

Rozaida Ghazali, Abir Jaafar Hussain, Dhiya Al-Jumeily, and Madjid Merabti

School of Computing & Mathematical Sciences,  
Liverpool John Moores University, L3 3AF Liverpool, England  
{cmprghaz, a.hussain, d.aljumeily, M.Merabti}@livjm.ac.uk

**Abstract.** This paper proposed a novel dynamic system which utilizes Ridge Polynomial Neural Networks for the prediction of the exchange rate time series. We performed a set of simulations covering three uni-variate exchange rate signals which are; the JP/EU, JP/UK, and JP/US time series. The forecasting performance of the novel Dynamic Ridge Polynomial Neural Network is compared with the performance of the Multilayer Perceptron and the feedforward Ridge Polynomial Neural Network. The simulation results indicated that the proposed network demonstrated advantages in capturing noisy movement in the exchange rate signals with a higher profit return.

## 1 Introduction

Multilayer Perceptrons (MLPs), the widely reported models have been effectively applied in financial time series forecasting. However, MLPs utilize computationally intensive training algorithms such as the error back-propagation and can get stuck in local minima. The networks also cannot elude the problem of slow learning, especially when they are used to solve complex nonlinear problems [1].

In contrast to the MLPs, Ridge Polynomial Neural Networks (RPNNs) [2] are simple in their architectures and they have only one layer of trainable weights; therefore they require less number of weights to learn the underlying equation. As a result, they can learn faster since each iteration of the training procedure takes less time [3], [4]. This makes them suitable for complex problem solving where the ability to retrain or adapt to new data in real time is critical. RPNNs have become valuable computational tools in their own right for various tasks such as pattern recognition [5], function approximation [2], time series prediction [4], and system control [3].

MLPs and RPNNs which are feedforward networks can only implement a static mapping of the input vectors. In order to model dynamical functions of the brain, it is essential to utilize a system that is capable of storing internal states and implementing complex dynamics. Since the behavior of the financial signal itself related to some past inputs on which the present inputs depends, the introduction of recurrence feedback in a network will lead to a proper input-output mapping. As a result, in this paper, we extend the functionality and architecture of feedforward RPNN by introducing a feedback from the output layer to the input layer in order to represent a dynamic system for financial time series prediction. Three uni-variate exchange rate

signals were used to test the performance of the networks which are the exchange rate between the Japanese Yen to US Dollar (JP/US), the Japanese Yen to Euro (JP/EU), and the Japanese Yen to UK Pound (JP/UK). The application of the novel Dynamic Ridge Polynomial Neural Network to financial time series prediction showed improvement in the annualized return and correct directional change in comparison to the MLP and the RPNN models.

## 2 The Networks

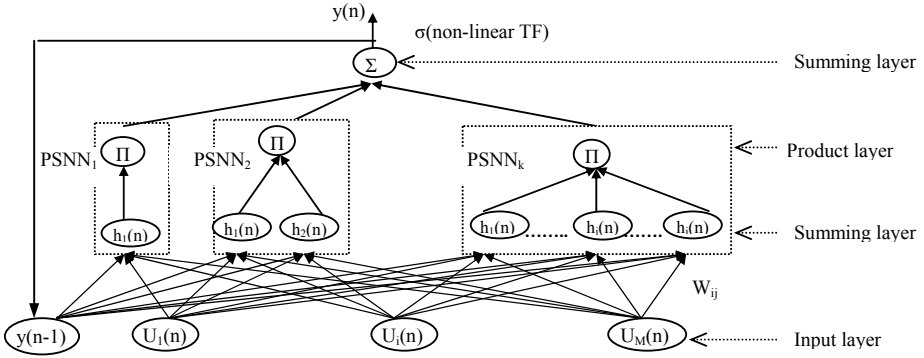
In this section, we describe the networks used in this work, addressing their architectures and capabilities in mapping the input and output patterns.

### 2.1 Ridge Polynomial Neural Network (RPNN)

RPNN was introduced by Shin and Ghosh [2]. The network has a single layer of adaptive weights. It has a well regulated High Order Neural Network (HONN) structure which is achieved through the embedding of different degrees of Pi-Sigma Neural Networks (PSNNs) [6]. RPNN provides a natural mechanism for incremental network growth which allows them to automatically determine the necessary network order, in order to carry out the required task. The network can approximate any multivariate continuous functions on a compact set in multidimensional input space, with arbitrary degree of accuracy [2]. Contrary to the ordinary HONN which utilizes multivariate polynomials, thus leading to an explosion of weights, RPNN uses univariate polynomials which are easy to handle. RPNN provides efficient and regular structure in comparison to ordinary HONN and the network has the ability to transform the nonlinear input space into higher dimensional space where linear separability is possible [7]. High order terms in the RPNN can increase the information capacity of the network and can help solving complex problems with construction of significantly smaller network while maintaining fast learning capabilities [4], [5].

### 2.2 Dynamic Ridge Polynomial Neural Network (DRPNN)

Neural networks with recurrent connections are dynamical systems with temporal state representations. They have been successfully used for solving varieties of problems [8], [9]. Motivated by the ability of the recurrent dynamic systems in real world applications, this paper proposes a Dynamic Ridge Polynomial Neural Network (DRPNN) architecture. The structure of the proposed DRPNN is constructed from a number of increasing order Pi-Sigma units with the addition of a feedback connection from the output layer to the input layer. The feedback connection feeds the activation of the output node to the summing nodes in each Pi-Sigma units, thus allowing each building block of Pi-Sigma unit to see the resulting output of the previous patterns. In contrast to RPNN, the proposed DRPNN is provided with memories which give the network the ability of retaining information to be used later. All the connection weights from the input layer to the first summing layer are learnable, while the rest are fixed to unity. Figure 1 shows the structure of the proposed DRPNN.



**Fig. 1.** Dynamic Ridge Polynomial Neural Network of  $k$ -th order

Suppose that we have  $M$  number of external inputs  $U(n)$  to the network, and let  $y(n-1)$  to be the output of the DRPNN at previous time step. The overall inputs to the network is the concatenation of  $U(n)$  and  $y(n-1)$ , and is referred to as  $Z(n)$  where:

$$Z_i(n) = \begin{cases} U_i(n) & \text{if } 1 \leq i \leq M \\ y(n-1) & i = M + 1 \end{cases} \quad (1)$$

The output of the  $k_{th}$  order DRPNN is determined as follows:

$$y(n) = \sigma \sum_{i=1}^k P_i(n)$$

$$P_i(n) = \prod_{j=1}^i (h_j(n)) \quad . \quad (2)$$

$$h_j(n) = \sum_{i=1}^{M+1} W_{ij} Z_i(n) + W_{j0}$$

where  $k$  is the number of Pi-Sigma units used,  $P_i(n)$  is the output of each PSNN block,  $h_j(n)$  is the net sum of the sigma unit in the corresponding PSNN block,  $W_{jo}$  is the bias, and  $\sigma$  is the sigmoid activation function.

### 3 Learning Algorithm for the Proposed Network

The DRPNN uses a constructive learning algorithm based on the asynchronous updating rule of the Pi-Sigma unit. The network adds a Pi-Sigma unit of increasing order to its structure when the difference between the current and the previous errors is less than a predefined threshold value. DRPNN follows the following steps for updating its weights:

1. Start with low order DRPNN
2. Carry out the training and update the weights asynchronously after each training pattern.
3. When the observed change in error falls below the predefined threshold  $r$ , i.e.,
 
$$\left| \frac{(e(n) - e(n-1))}{e(n-1)} \right| < r$$
, a higher order PSNN is added.
4. The threshold  $r$ , for the error gradient together with the learning rate  $n$ , are reduced by a suitable factor  $dec\_r$  and  $dec\_n$ , respectively.
5. The updated network carries out the learning cycle (repeat steps 1 to 4) until the maximum number of epoch is reached.

Notice that every time a higher order PSNN is added, the weights of the previously trained PSNN networks are kept frozen, whilst the weights of the latest added PSNN are trained. The weights of the pi-sigma units are updated using the real time recurrent learning algorithm [10]. In this work, a standard error measure used for training the network is the Sum Squared Error:

$$E(n) = \frac{1}{2} \sum e(n)^2 \tag{3}$$

The error between the target and actual signal is determined as follows:

$$e(n) = d(n) - y(n), \tag{4}$$

where  $d(n)$  is the target output at time  $n$ ,  $y(n)$  is the forecast output at time  $n$ . At every time  $n$ , the weights are updated according to:

$$\Delta W_{kl}(n) = -\eta \frac{\partial E(n)}{\partial W_{kl}}, \tag{5}$$

where  $\eta$  is the learning rate. The value  $\left( \frac{\partial E(n)}{\partial W_{kl}} \right)$  is determined as:

$$\left( \frac{\partial E(n)}{\partial W_{kl}} \right) = e(n) \frac{\partial y(n)}{\partial W_{kl}}. \tag{6}$$

$$\frac{\partial y(n)}{\partial W_{kl}} = \frac{\partial y(n)}{\partial P_i(n)} \frac{\partial P_i(n)}{\partial W_{kl}}, \tag{7}$$

where

$$\frac{\partial y(n)}{\partial P_i(n)} = f \left( \sum_{i=1}^k P_i(n) \right) \left( \prod_{\substack{j=1 \\ j \neq i}}^i h_j(n) \right) \tag{8}$$

and

$$\frac{\partial P_i(n)}{\partial W_{kl}} = \left( W_{ij} \frac{\partial y(n-1)}{W_{kl}} \right) + Z_j(n) \delta_{ik}, \quad (9)$$

where  $\delta_{ik}$  is the Krocnoker delta. Assume  $D$  as the dynamic system variable, where  $D$  is:

$$D_{ij}(n) = \frac{\partial y(n)}{\partial W_{kl}}. \quad (10)$$

Substituting equation (8) and (9) into (7) results in:

$$D_{ij}(n) = \frac{\partial y(n)}{\partial W_{kl}} = f' \left( \sum_{i=1}^k P_i(n) \right) \times \left( \prod_{\substack{j=1 \\ j \neq i}}^i h_j(n) \right) \left( W_{ij} D_{ij}(n-1) + Z_j(n) \delta_{ik} \right), \quad (11)$$

where initial values for  $D_{ij}(n-1)=0$ , and  $Z_j(n)=0$ . Then the weights updating rule is:

$$\begin{aligned} \Delta W_{ij}(n) &= \eta e(n) D_{ij}(n) + \alpha \Delta W_{ij}(n-1) \\ W_{ij}(n+1) &= W_{ij}(n) + \Delta W_{ij}(n) \end{aligned} \quad (12)$$

## 4 Financial Time Series

Three daily exchange rate signals are considered in this paper; the JP/EU, JP/US, and JP/UK exchange rates. All signals were obtained from a historical database provided by DataStream, dated from 03/01/2000 until 04/11/2005, giving a total of 1525 data points. To smooth out the noise and to reduce the trend, the non-stationary raw data was pre-processed into stationary series by transforming them into 5-day relative different in percentage of price (RDP) [11]. The advantage of this transformation is that the distribution of the transformed data will become more symmetrical and will follow more closely to normal distribution. This means that most of the transformed data are close to the average value, while relatively few data tend to one extreme or the other.

The input variables were determined from 4 lagged RDP values based on 5-day periods (RDP-5, RDP-10, RDP-15, and RDP-20) and one transformed signal (EMA15) which is obtained by subtracting a 15-day exponential moving average from the original signal. The calculations for the transformation of input and output variables are presented in Table 1.

As mentioned in [11], the optimal length of the moving day is not critical, but it should be longer than the forecasting horizon. Since the use of RDP to transform the original series may remove some useful information embedded in the data, EMA15 was used to retain the information contained in the original data. Smoothing both input and output data by using either simple or exponential moving average is a good



**Table 1.** Calculations for input output variables

	Indicator	Calculations
Input variables	EMA15	$P(i) - EMA_{15}(i)$
	RDP-5	$(p(i) - p(i - 5)) / p(i - 5) * 100$
	RDP-10	$(p(i) - p(i - 10)) / p(i - 10) * 100$
	RDP-15	$(p(i) - p(i - 15)) / p(i - 15) * 100$
	RDP-20	$(p(i) - p(i - 20)) / p(i - 20) * 100$
Output variable	RDP+5	$\frac{(p(i + 5) - p(i)) / p(i) * 100}{p(i) = EMA_3(i)}$

approach and can generally enhance the prediction performance [12]. The weighting factor,  $\alpha=[0,1]$  determines the impact of past returns on the actual volatility. The larger the value of  $\alpha$ , the stronger the impact and the longer the memory. In our work, exponential moving average with weighting factor of  $\alpha=0.85$  was experimentally selected.

From the trading aspect, the forecasting horizon should be sufficiently long such that excessive transaction cost resulted from over-trading could be avoided [13]. Meanwhile, from the prediction aspect, the forecasting horizon should be short enough as the persistence of financial time series is of limited duration. Thomason [11] suggested that a forecasting horizon of five days is a suitable choice for the daily data. Therefore, in this work, we consider the prediction of a relative different in percentage of price for the next five business day. The output variable, RDP+5, was obtained by first smoothing the signal with a 3-day exponential moving average, and is presented as a relative different in percentage of price for five days ahead. Because statistical information of the previous 20 trading days was used for the definition of the input vector, the original series has been transformed and is reduced by 20. The input and output series were subsequently scaled using standard minimum and maximum normalization method which then produces a new bounded dataset. One of the reasons for using data scaling is to process outliers, which consist of sample values that occur outside normal range.

## 5 Simulation Results and Discussion

The main interest in financial time series forecasting is how the networks generate profits. Therefore, during generalization, the network model that endows the highest profit on unseen data is considered the best model. The networks ability as traders was evaluated by the Annualized Return (AR) [14], where the objective is to use the networks predictions to make money. Two statistical metrics; the Correct Directional Change (CDC) [14] and the Normalized Mean Squared Error (NMSE) were used to provide accurate tracking of the signals. For all neural networks, an average performance of 20 trials was used and the network parameters were experimentally selected as shown in Table 2. A sigmoid activation function was employed and all networks were

trained with a maximum of 3000 epochs. MLPs and RPNNs were trained with the incremental backpropagation algorithm [15] and incremental learning algorithm [2], respectively. We trained the DRPNNs with the learning algorithm as described in Sect. 3. The higher order terms of the RPNNs and the DRPNNs were selected between 2 to 5.

**Table 2.** The learning parameters used for DRPNNs and the benchmarked networks

Neural Networks	Initial Weights	Learning Rate (n)	dec_n	Threshold (r)	dec_r
MLP	[-0.5,0.5]	0.1 or 0.05	-	-	-
RPNN & DRPNN	[-0.5,0.5]	[0.1, 0.5]	0.8 or 09	[0.001,0.7]	[0.05,0.2]

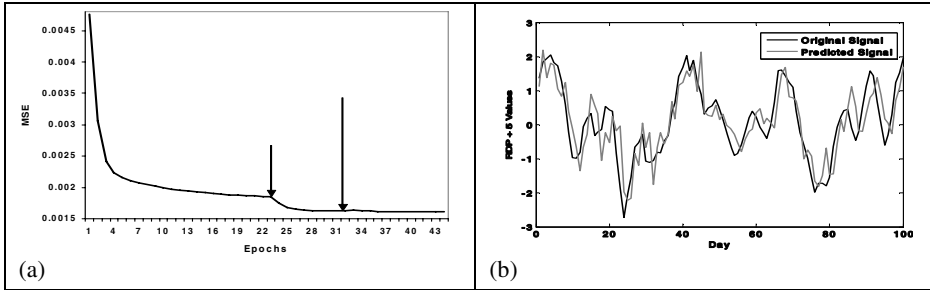
The best average results achieved on unseen data is summarized in Table 3. The results apparently show that the proposed DRPNNs outperformed the MLPs and RPNNs in terms of attaining the best profit on average for all signals. The profit made by DRPNNs is about 0.16% to 2.65% greater than that of other neural networks architectures. MLPs in this case, clearly appeared to produce the lowest profit for all signals. In terms of predicting the subsequent actual change of a forecast variable, DRPNNs reached the highest CDC for JP/US and JP/UK signals, which indicates that the network predicted the directional symmetry of the two signals more accurately than MLPs and RPNNs. Meanwhile, the prediction error of DRPNNs is slightly higher than that of other models. Albeit higher NMSE, the performance of DRPNNs is considered to be satisfactory with respect to the high profit generated by the networks. It is worth noting that seeking a perfect forecasting in terms of the prediction error is not our aim, as we are more concern with the out-of-sample profitability.

In the case of learning speed, DRPNNs converged much faster than MLPs which is about 2 to 99.8 times faster when used to learn all the signals. However, DRPNNs used more epochs than RPNNs for the prediction of JP/UK and JP/EU signals. This was owing to the large number of free parameters employed by the DRPNN. Figure 2(a) shows the learning curve for the prediction of JP/EU using DRPNN, which demonstrates that the network has the ability to learn the signals very quickly. The learning was quite stable and the MSE decreased drastically when a 2<sup>nd</sup> order PSNN is added to the network.

**Table 3.** The average performance of the DRPNNs and the benchmarked models

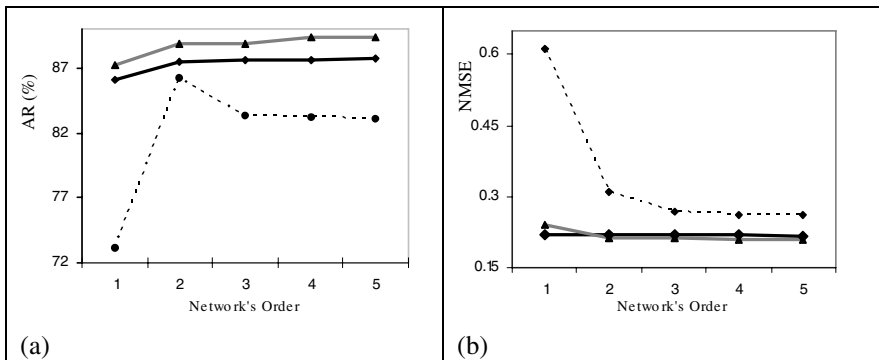
JP/US		AR (%)	CDC	NMSE	Epoch
MLP	- Hidden 6	83.55	58.49	0.2694	699
RPNN	- Order 2	84.84	58.59	0.2927	8
DRPNN	- Order 2	86.20	59.67	0.3114	7
JP/UK		AR (%)	CDC	NMSE	Epoch
MLP	- Hidden 5	88.97	59.51	0.2083	1179
RPNN	- Order 4	89.25	59.68	0.2084	298
DRPNN	- Order 4	89.41	60.41	0.2090	596
JP/EU		AR (%)	CDC	NMSE	Epoch
MLP	- Hidden 7	87.05	64.69	0.2156	3000
RPNN	- Order 4	87.48	64.24	0.2152	817
DRPNN	- Order 5	87.75	64.56	0.2168	945

For the purpose of demonstration, the best forecast generated by the DRPNN when used to predict the JP/EU signal is depicted in Fig. 2(b). As it can be noticed from the plot, the DRPNN is capable of learning the behaviour of chaotic and highly non-linear financial time series and they can capture the underlying movements in financial markets.



**Fig. 2.** (a) Learning curve for the prediction of JP/EU using DRPNN, (b) Best forecast made by DRPNN on JP/EU; — is original signal, and - - - is predicted signal

In order to test the modelling capabilities and the stabilities of DRPNNs, Fig. 3(a) and Fig. 3(b) show the average result of AR and NMSE tested on out-of-sample data when used to predict all the signals. The performance of the networks was evaluated with the number of higher order terms increased from 1 to 5. The plots indicated that the DRPNNs learned the data steadily with the AR continues to increase, while the NMSE keeps decreasing along with the network growth. For the prediction of JP/US signal, the percentage of AR started to decrease when a 3<sup>rd</sup> order PSNN unit is added to the network. This is probably due to the utilization of large number of free parameters for the network of order three and more has led to unpromising generalization for the input-output mapping of that particular signal.



**Fig. 3.** Performance of DRPNN with increasing order; .....JP/US, — JP/UK, — JP/EU

DRPNN generalized well and achieved high annualized return which is a desirable property in nonlinear financial time series prediction. The network learned the underlying mapping steadily as the order of the network increased. This indicates that the interaction between the input signals of DRPNN of order two to five contains significant information for the prediction task. Our simulation results indicated that the DRPNN is a suitable modelling tool for the prediction of financial time series signals since it incorporates the supplementary information from the feedback output which acts as an additional guidance to evaluate the current noisy input and its signal component.

## 6 Conclusion

In this paper, we proposed a novel Dynamic Ridge Polynomial Neural Network to forecast the next 5-day relative different price of 3 daily exchange rate signals, and benchmarking it against the performance of the Ridge Polynomial Neural Network and the Multilayer Perceptron. Results showed that a significant profitable value does exist in the DRPNN when compared to the benchmarked networks. The general property making the DRPNN interesting and such potentially useful in financial prediction is that it manifests highly nonlinear dynamical behaviour induced by the recurrent feedback, therefore leads to a better input-output mapping.

## References

1. Chen, A.S., Leung, M.T.: Regression Neural Network for Error Correction in Foreign Exchange Forecasting and Trading. *Computers & Operations Research*, 31 (2004) 1049-1068
2. Shin, Y., Ghosh, J.: Ridge Polynomial Networks. *IEEE Transactions on Neural Networks*, Vol.6, No.3, (1995) 610-622
3. Karnavas, Y.L., Papadopoulos, D.P.: Excitation Control of a Synchronous Machine using Polynomial Neural Networks. *Journal of ELECTRICAL ENGINEERING*, Vol. 55, No. 7-8, (2004) 169-179
4. Tawfik, H., Liatsis, P.: Prediction of Non-linear Time-Series using Higher-Order Neural Networks. *Proceeding IWSSIP'97 Conference*, Poznan, Poland, (1997)
5. Voutriaridis, C., Boutalis, Y.S., Mertzios, G.: Ridge Polynomial Networks in Pattern Recognition. *EC-VIP-MC 2003, 4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications*, Croatia. (2003) 519-524
6. Shin, Y., Ghosh, J.: The Pi-Sigma Networks: An efficient Higher-Order Neural Network for Pattern Classification and Function Approximation. *Proceedings of International Joint Conference on Neural Networks*, Vol.1, Seattle, Washington (1991) 13-18
7. Pao, Y.: *Adaptive Pattern Recognition and Neural Networks*. Addison Wesley, (1989)
8. Yumlu, S., Gurgun, F.S., Okay, N.: A Comparison of Global, Recurrent and Smoothed-Piecewise Neural Models for Istanbul Stock Exchange (ISE) Prediction. *Pattern Recognition Letters* 26 (2005) 2093-2103
9. Medsker, L.R., Jain, L.C.: *Recurrent Neural Networks: Design and Applications*. CRC Press LLC, USA, ISBN 0-8493-7181-3. (2000)
10. Williams, R.J., Zipser, D.: A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1 (1989) 270-280

11. Thomason, M.: The Practitioner Method and Tools, *Journal of Computational Intelligence in Finance*. Vol. 7, no. 3 (1999) 36-45
12. Thomason, M.: The Practitioner Method and Tools, *Journal of Computational Intelligence in Finance*. Vol. 7, no. 4 (1999) 35-45
13. Cao, L. J., Francis E. H. T.: Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting, *IEEE Transactions on Neural Networks*, Vol. 14, no. 6 (2003) 1506-1518
14. Dunis, C.L., Williams, M.: Modeling and Trading the UER/USD Exchange Rate: Do Neural Network Models Perform Better?. in *Derivatives Use, Trading and Regulation*, 8 (3) (2002) 211-239
15. Haykin, S.: *Neural Networks. A comprehensive Foundation*. Second Edition, Prentice-Hall, Inc., New Jersey, (1999)

# Neural Systems for Short-Term Forecasting of Electric Power Load

Michał Bąk and Andrzej Bielecki

Institute of Computer Science, Jagiellonian University,  
Nawojki 11, 30-072 Kraków, Poland

michal.bak@westernconsulting.co.uk, bielecki@softlab.ii.uj.edu.pl

**Abstract.** In this paper a neural system for daily forecasting of electric power load in Poland is presented. Basing on the simplest neural architecture - a multi-layer perceptron - more and more complex system is built step by step. A committee rule-aided hierarchical system consisting of modular ANNs is obtained as a result. The forecasting mean absolute percentage error (MAPE) of the most effective system is about 1.1%.

## 1 Introduction

Methods of precise forecasting of electric power load are in constant demands on electricity markets. This is caused by the specific character of electric power which, particularly, can not be stored and therefore balance between demand and supply must be managed in real time ([23], page 53). Statistical methods, time series and artificial intelligence systems (AI systems), including expert systems, fuzzy systems, abductive networks, artificial neural networks (ANNs) and hybrid systems, are used to perform this task ([1], [2], [4], [5], [6], [9], [12], [14] - Sect. 4.5, 8.5, [17], [19], [20], [24] - Sect. 3.7, 4.6, 5.6, 8.2). Classical methods - statistical ones and time series - turn out to be insufficient because of dynamical changing of electric load. Therefore AI systems are widely used for predicting of power request because of their efficiency in solving of prediction problems ([8], [15]).

Short term forecasting, particularly daily prediction of power demand, is a specific problem concerning electric markets. In this context the possibility of electric power load forecasting for various countries was tested ([1], [2], [9], [10], [11], [19]). In AI systems the mean-absolute percentage error (MAPE) for daily forecasts varies from 1.5% to 3.5% according to the type of a system and concrete approach ([3], [5], [16], [18], [24] - pages 50, 138 - Table 4.10, 178 - Table 5.4).

In this paper a specific approach for a daily forecasting of electric power load in Poland, basing on ANNs, is presented. Basing on the simplest neural architecture - a multi-layer perceptron - more and more complex system is built step by step. A committee rule-aided hierarchical system consisting of modular ANNs is obtained as a result. Results obtained by other authors are described in Sect. 2 whereas in Sect. 3 the proposed system is put. It should be stressed that results for systems described in Sects. 2.3 and 2.4 are introductory ones.

## 2 AI Systems for Short-Term Forecasting of Electric Power Load

Implemented neural systems for daily forecasting of electric power load are based on a few approaches. A multi-layer perceptron is one of the most widely used neural system. It was used for one-hour prediction of power load ([14] - Sect. 4.5). The mathematical model is postulated of the form

$$p(i, t) = f(\mathbf{W}, y(i, t-1), y(i-1, t), \dots, y(i-1, t-k), y(i-2, t), \dots, y(i-d, t-k)),$$

where  $\mathbf{W}$  is the vector of all weights of the ANN,  $y(m, \tau)$  is a known load on the  $m$ th day at  $\tau$ th hour,  $d$  and  $k$  denotes a number of previous days and hours taken into account. Components of the input vector represented a type of the day of a week (there were distinguished four sorts of days: Saturday, Sunday, Monday and other workdays), load at the  $t$ th and  $(t-1)$ th hour for the  $(i-1)$ th,  $(i-2)$ th and  $(i-3)$ th days and at  $(t-1)$ th hour of the  $i$ th day when the load on the  $i$ th day at  $t$ th hour was predicted. The network structure  $8-15-10-1$  turned out to be the most effective one. For this ANN the MAPE =  $\frac{100}{N} \sum_{n=1}^N \frac{|y_n - p_n|}{y_n}$ , where  $p$  is a predicted value and  $y$  is a measured one, was equal to 1.29%.

A little different approach, also basing on a multi-layer perceptron, is described in [24], Sect. 5.6. The mathematical model is following

$$D(k) = g(G(1, k-1), \dots, G(24, k-1), T_n(k-1), T_x(k-1), T_n(k), T_x(k), d), \quad (1)$$

where  $D(k)$  denotes the predicted total power load on the day number  $k$ ,  $G(i, k-1)$ ,  $i \in \{1, \dots, 24\}$  is the distribution of the load on the previous day,  $T_n(k-1)$  and  $T_x(k-1)$  is a maximal and minimal temperature on the previous day respectively,  $T_n(k)$  and  $T_x(k)$  are predicted temperatures for the day for which the forecasting is done and  $d$ , coding the type of the day, is equal to 0 if the day  $k$  is Sunday and 1 otherwise. Such set of input variables was selected basing on statistical analysis of correlations coefficients - see [12]. The MAPE of the most effective perceptron was equal to 2.04% whereas MAPE of linear regression was 2.39%. The fuzzy system implemented as a neural-like system, basing also on the mathematical model (1) gave a little better results - MAPE = 1.99% - see [24], Table 5.4, page 178.

In the thesis [21] three approaches to power load forecasting are presented - the daily load prediction was done using a perceptron, Kohonen network and fuzzy system neurally aided. The load at previous hours, type of the day (four types - the same as in the system described in [14]), part of the day according to the power load (four possibilities: night minimum, morning maximum, afternoon minimum and evening maximum) and season of the year were put as the perceptron input vector. The MAPE of the best perceptron was 2.99% for a daily forecasting. Using Kohonen network similar daily power load trends are grouped into clusters. Forecasting is done by averaging weights vector of neurons representing clusters to which the type of the predicted day belongs. The MAPE for the best Kohonen network was 2.58% whereas the MAPE of a fuzzy predicting system was 2.51%. In other papers ([9], [14] - sect. 8.5) there are described Kohonen networks for which MAPE is equal to 1.5%.

A modular approach in which a committee (ensemble) of neural networks, each trained on the data of one year, is described in the paper [4]. Combining the outputs of such networks can improve the accuracy and reliability of the forecasts beyond those of individual networks and of the single monolithic model developed using the full data for all years. The MAPE of the presented system was equal to 2.61%.

### 3 A Hierarchical Hybrid Forecasting System

In this section neural systems for 24-hour forecasting of electric power load for Poland are described. Data stored by Polish Power Grid covered the time interval from 1st January 2000 to 30th April 2004 and included information about power load every fifteen minutes. Introductory data analysis shown that average annual power load is practically constant. A few types of cycles can be observed at the power load trend: 24-hour one, weekly and annual but the 24-hour cycle is not exactly repeated. It turns out that there are four types of daily power load trends: for Saturdays, Sundays, Mondays and the last one for other workdays (see the previous section). Monday power load differs from other workdays for first seven hours.

The hybrid hierarchical system was built in four steps according to methodology presented in [7] which means, among others, that the system proposed in the  $i$ th step became a part of the system implemented in the  $(i + 1)$ th step. The single multi-layer perceptron (Sect. 3.1) was the simplest system. In the best configuration its MAPE was equal to about 4%. The modular network consisting of the perceptrons having the same architecture as the best one obtained in the first step was the second level of the system complexity (Sect. 3.2). This is a classical example of parallel modular neural system [13]. The lowest MAPE of this kind of the system was equal to above 2%. The committee system of modular networks is the third level of the system complexity and the MAPE of the best system was lower than 2%. The system gave good predictions for workdays and ordinary weekends but for untypical holidays, for instance New Year, predictions was significantly worse. The little number of examples for such holidays was one of the reasons of such situation. In order to improve the load prediction in untypical days a hybrid rule-neural system was implemented. For typical days the forecast was done by the committee neural system and for extra holidays the prediction was done using rules. This system achieved the highest accuracy - MAPE of prediction was reduced to about 1% which means that the system is better than any one described in literature - see Sect. 2.

#### 3.1 Single Multi-layer Neural Network

Multi-layer perceptron was the simplest neural system used for electric load twenty-four hour forecasting. Every tested network had the following architecture: thirteen-component input, one hidden layer with sigmoidal neurons and one output neuron. The input vector had the following components.



1. Power load at the three previous hours  $L(i - 1), L(i - 2), L(i - 3)$ .
2. Power load on the previous day at the same hour and for neighbouring ones:

$$L(i - 22), L(i - 23), L(i - 24), L(i - 25), L(i - 26).$$

3. Mean temperature at last three hours:  $(T(i - 1) + T(i - 2) + T(i - 3))/3$ .
4. Mean temperature on the previous day at the same hour and for neighbouring hours

$$(T(i - 22) + T(i - 23) + T(i - 24) + T(i - 25) + T(i - 26))/5.$$

5. The number of the day in week (1..7) for which the forecasting is done.
6. The number of the day in year (1..366).
7. The hour for which the forecasting is done.

The input data were normalized in such a way that temperature belonged to the interval  $[-1, 1]$  and other components to  $[0, 1]$ . In a twenty-hour forecast only for the starting hour  $i$  the loads at three previous hours  $L(i - 1), L(i - 2), L(i - 3)$  are genuine ones. Predicting the loads for other hours the corresponding loads in the input vector were equal to values which had been predicted. Thus in the twenty-hour forecasting the error cumulated. Various numbers of hidden neurons and learning epochs were tested - see Table 1. Perceptrons were trained using the Levenberg-Marquardt algorithm which has good convergent properties. It turned out that about twenty training epochs is the optimal length of learning process. The forecasting accuracy (MAPE) changed from about 3% to 7% - see Tables 1 and 2. It can be observed that the MAPE of Sundays is far greater than for other days.

**Table 1.** MAPE of twenty-hour forecasting for various days of a week and various number of learning epochs for the perceptrons with 12 and 25 hidden neurons. One o'clock was the starting hour.

12 neurons in the hidden layer								
Epochs	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Mean
10	15.21	5.63	3.47	2.70	3.17	4.45	5.36	5.71
20	18.03	6.91	3.03	2.55	3.44	3.52	2.52	5.71
50	24.69	8.00	3.38	2.70	3.48	3.27	2.79	6.90
25 neurons in the hidden layer								
Epochs	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Mean
10	8.72	5.96	2.23	2.86	2.49	2.84	3.27	4.05
20	16.70	5.17	2.10	2.69	2.40	2.69	4.54	5.18

### 3.2 Modular Neural System

In order to improve the forecasting a specific modular system was proposed. The system consisted of twenty four perceptrons and each of them predicted a load at different hour. Every network of the system had the same architecture - one hidden layer with sigmoidal neurons and one output neuron.

**Table 2.** MAPE of twenty-hour forecasting for various starting hours for the perceptron with 25 hidden neurons learnt by 20 epochs

Hour	MAPE	Hour	MAPE	HOUR	MAPE	HOUR	MAPE
1.00	5.18	7.00	3.76	13.00	3.47	19.00	4.52
2.00	4.45	8.00	3.42	14.00	3.52	20.00	5.01
3.00	3.92	9.00	3.67	15.00	3.77	21.00	5.04
4.00	3.71	10.00	3.18	16.00	3.97	22.00	5.21
5.00	3.63	11.00	3.31	17.00	4.15	23.00	5.62
6.00	3.72	12.00	3.44	18.00	4.19	24.00	5.14

Assuming that the system had to predict the power load on a given day at the  $i$ th hour the data vector was put onto the input of the  $i$ th (network denoted by  $S_i$ ) of the system. The input vector had twelve components - the same as the input vector of the single network apart from the last one coding the hour the prediction is done for.

A twenty-hour forecasting was done step by step in the following way. An input vector was given to the network  $S_i$  which predicted the load at the  $i$ th hour - denote it by  $P_i$ . Then in the input vector for the  $S_{i+1}$  network  $L((i+1)-1) = P_i$  and the predicted value  $P_{i+1}$  were calculated. In the input vector for the  $S_{i+2}$  network  $L((i+1)-1) = P_{i+1}$  and  $L((i+1)-2) = P_i$  etc. The obtained vector  $[P_i, \dots, P_{i+23}]$  was a twenty-hour electric power load forecasting. Accuracy of forecasting for modular systems having various numbers of neurons in a hidden layer are presented in Table 3. Predictions were done for a testing set different from the learning one. MAPE was calculated every day of the testing set and then averaged.

**Table 3.** MAPE of twenty-hour forecasting for various numbers of neurons in hidden layer averaged of starting hours

Number of neurons	5	9	12	16
MAPE	2.13	2.09	2.06	2.07

All presented so far results were calculated assuming that one o'clock was the starting hour. However it turned out that the forecasting error depended on the starting hour. The results of simulations are shown in Table 4 for various number of learning epochs. For the same starting hour the initial weights were the same for every number of epochs.

In Table 5 results of twenty-four hour forecasting for each day of week are presented for one hundred learning epochs and for eleven o'clock as the starting hour. It can be observed that the forecasting was most efficient if eleven o'clock was the starting hour MAPE = 1.5% - see Table 4.

**Table 4.** MAPE of twenty-hour forecasting for various starting hours

Starting hour	Number of epochs			Starting hour	Number of epochs		
	20	50	100		20	50	100
1	2.22	2.13	2.09	13	1.62	1.62	1.66
2	2.22	2.17	2.08	14	1.72	1.71	1.75
3	2.23	2.19	2.09	15	1.82	1.81	1.85
4	2.17	2.14	2.05	16	1.93	1.93	1.96
5	2.09	2.07	1.99	17	2.04	2.05	2.06
6	2.06	2.04	1.97	18	2.14	2.14	2.13
7	2.02	2.01	1.91	19	2.18	2.20	2.20
8	1.69	1.67	1.69	20	2.24	2.29	2.26
9	1.63	1.60	1.63	21	2.23	2.30	2.30
10	1.55	1.53	1.58	22	2.25	2.32	2.31
11	1.51	1.50	1.55	23	2.23	2.24	2.26
12	1.56	1.55	1.61	24	2.17	2.12	2.07

**Table 5.** MAPE of twenty-hour forecasting for each day of week, eleven o'clock is the starting hour

Day	MAPE
Monday	1.32
Tuesday	1.34
Wednesday	1.69
Thursday	1.57
Friday	1.48
Saturday	1.77
Sunday	1.67
Mean MAPE	1.55

### 3.3 Committee Neural System

Committee machine ([22]) is a system of neural networks such that every one is trained for solving the same task. Such systems were also used for short term forecasting of electric power load ([4]). Calculating the answer of the system results obtained by all networks are taken into account. Usually they are averaged. In order to improve a forecasting a committee of modular neural systems described in the previous subsection was implemented. Let  $K$  denotes a number of modular systems the committee consists of. Let, furthermore,  $S_{ik}, i = 1, \dots, 24, k = 1, \dots, K$  be the  $i$ th neural network in the  $k$ th module, i.e. the network predicting power load at  $i$ th hour in the  $k$ th module and let  $P_k(i)$  denotes forecasting of the  $i$ th network in the  $k$ th committee. Then the forecasting of the power load  $P(i)$  at the  $i$ th hour of a given day is a mean value of predictions of single networks  $P(i) = \frac{1}{K} \sum_{k=1}^K P_k(i)$ . The obtained results are

introductory ones but it can be observed that application of committees can significantly improve the forecasting effectiveness.

Committees consisted of various number of modular systems were tested -  $K$  varied from 2 to 30. A learning sequence was given to every single network one hundred times (100 learning epochs) and was trained in the way described in the previous subsection. The only difference was that predicting power load later than at the starting hour the predicted value calculated by the whole system (not by a single network) was given as a respect component of the input vector. Introductory simulations shown that the MAPE decreases if  $k$  increases until  $k$  reaches value equal to five. Then the MAPE remains constant. For every starting hour the MAPE of the committee is significantly less than for single networks the committee consists of. For the best committee system MAPE was about 1.3% - the results are introductory.

### 3.4 Rule-Aided Neural System

There are fourteen extra holidays in Poland - see Table 6. Data analysis shown that they generate six untypical twenty-hours power load trends:

1. Extra holidays;
2. Monday before an extra holiday;
3. Tuesday, Wednesday and Thursday after an extra holiday;
4. Friday after an extra holiday;
5. Saturday after an extra holiday;
6. Saturday two days after an extra holiday.

For all the listed sorts of untypical days the MAPE is significantly greater than for both workdays and ordinary weekends because not only of specific character of their power load trends but of low level of power load as well.

For every extra holiday, basing on learning data, its typical daily power load trend had been calculated by data averaging. Then, the obtained trends were glued to the power load trend curve of the previous day. In such a way the proper starting levels for the trend of untypical days were obtained. The accuracy of such prediction is shown in Table 6. The system was modified in such a way that extra holidays power load data was removed from the learning set of the neural committee system and if the time interval for which the forecasting is done has a common part with an extra holiday, then for hours of the holiday the prediction is taken as values of glued power load trend. The accuracy of the whole rule-neural system forecasting was reduced to about 1.1% (MAPE). It should be stressed that the obtained results are introductory - the rule system was done only for extra holidays i.e. only for untypical days of the type 1. However the system is developed permanently, particularly forecasting for untypical days of other types will done by the rule module in the next version of the system.

**Table 6.** The MAPE for extra holidays for two various starting hours

Holiday	One o'clock is the starting hour					Six o'clock is the starting hour				
	Year				Mean	Year				Mean
	2000	2001	2002	2003		2000	2001	2002	2003	
01 .01	5.19	1.48	1.66	2.48	2.70	2.61	1.85	1.36	0.74	1.64
01.05	6.27	1.11	2.64	2.61	3.16	5.50	0.98	3.08	2.55	3.03
02.05	2.14	1.48	1.11	1.04	1.44	1.80	0.91	1.06	0.98	1.19
03.05	1.81	5.17	1.02	4.34	3.08	1.88	2.76	1.16	4.13	2.48
15.08	4.22	1.27	1.83	1.33	2.16	2.22	0.69	1.24	1.10	1.31
01.11	2.19	2.44	1.57	2.86	2.27	1.27	0.85	0.78	1.15	1.01
11.11	4.09	3.71	5.98	1.77	3.89	1.52	3.21	4.37	0.64	2.43
24.12	4.13	3.49	1.10	1.54	2.56	3.39	1.26	1.31	1.79	1.94
25.12	0.88	1.30	0.98	0.74	0.98	0.67	1.33	1.16	0.81	0.99
26.12	0.77	1.20	0.96	1.17	1.02	1.01	0.62	1.38	0.86	0.97
31.12	8.01	3.71	1.80	2.80	4.08	4.64	1.09	1.28	3.16	2.54
Corpus Christi	3.85	2.17	1.71	4.23	2.99	1.00	2.27	1.34	2.03	1.66
Eastern Sunday	2.03	2.16	2.34	1.42	1.99	2.29	1.22	4.78	3.45	2.94
Eastern Monday	2.88	2.11	1.71	1.41	2.03	4.94	2.35	3.13	4.70	3.78

## 4 Concluding Remarks

In this paper not only systems for short-term electric power load forecasting are presented but specific properties of load trends are discussed as well. The power load dependence on the types of week days was well known previously but it also turned out that the efficiency of forecasting depends on starting hour. Data analysis also allowed to specify six sorts of untypical power trends connected with extra holidays (see Sect. 3.4) what was the starting point for designing the hybrid rule-neural system.

The obtained results allow to conclude that an iterative approach to AI systems implementation is effective. The prediction error of both modular and committee systems was as low as the most effective systems implemented by other authors whereas the error of forecasting in the hybrid rule-neural system was lower than the error of the best described systems. Furthermore, the results obtained for the committee and hybrid systems are introductory - in both these cases improved versions of the systems are tested. However it should be mentioned that the described methodology of an obtained predicting complex system is time consuming.

The obtained results are satisfactory (MAPE of about 1.1%) as they are comparable with the ones obtained by other authors in recent years. The paper [3], describing developing of 24 dedicated models for forecasting next-day hourly loads, can be put as example. Evaluated on data for the sixth year, the models gave an overall MAPE of 2.67%. Next-hour models utilizing available load data up to the forecasting hour gave a MAPE of 1.14%, outperforming neural network models for the same utility data.

## References

1. Abdel-Aal R.E., Al-Garni A.Z.: Forecasting monthly electric energy consumption in eastern Saudi Arabia using univariate time-series analysis. *Fuel and Energy Abstracts* **38** (1997) 452-452
2. Abdel-Aal R.E., Al-Garni A.Z., Al-Nassar Y.N.: Modelling and forecasting monthly electric energy consumption in eastern Saudi Arabia using abductive networks. *Energy* **22** (1997) 911-921
3. Abdel-Aal R.E.: Short-term hourly load forecasting using abductive networks. *IEEE Transactions on Power Systems* **19** (2004) 164-173
4. Abdel-Aal R.E.: Improving electric load forecasts using network committees. *Electric Power Systems Research* **74** (2005) 83-94
5. Abdel-Aal R.E.: Modeling and forecasting electric daily peak loads using abductive networks. *International Journal of Electrical Power and Energy Systems* **28** (2006) 133-141
6. Bartkiewicz W.: Confidence intervals prediction for the short-term electrical load neural forecasting models. *Elektrotechnik und Informationstechnik* **117** (2000) 8-12
7. Bielecki A., Bąk M.: Methodology of Neural Systems Development. In: Cader A., Rutkowski L., Tadeusiewicz R., Żurada J. (eds.): *Artificial Intelligence and Soft Computing. Challenging Problems of Science - Computer Science*. Academic Publishing House EXIT, Warszawa (2006) 1-7
8. Breiman L.: Bagging predictors. *Machine Learning*, **24** (1996) 123-140
9. Cottrell M., Girard B., Girard Y., Muller C., Rousset P.: Daily electrical power curve: classification and forecasting using a Kohonen map. In: Mira J., Sandoval F. (eds.): *From Natural to Artificial Neural Computation*. IWANN, Malaga (1995) 1107-1113
10. Djukanowic M., Babic B., Sobajic D.J., Pao Y.H.: Unsupervised/supervised learning concept for 24-hour load forecasting. *IEE Proceedings* **4** (1993) 311-318
11. Hsu Y.Y., Ho K.L.: Fuzzy expert systems: an application to short-term load forecasting. *IEE Proceedings*, **6** (1992) 471-477
12. Malko J.: *Certain Forecasting Problems in Electrical Power Engineering*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (1995) (in Polish)
13. Marciniak A., Korbicz J.: Modular neural networks. In: Duch W., Korbicz J., Rutkowski L., Tadeusiewicz R. (eds.): *Neural Networks, Biocybernetics and Biomedical Engineering*, Vol. 6. Academic Publishing House EXIT, Warszawa (2000) 135-178 (in Polish)
14. Osowski S.: *Neural Networks - an Algorithmic Approach*. WNT, Warszawa (1996) (in Polish).
15. Osowski S.: *Neural Networks for Information Processing*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa (2000) (in Polish).
16. Osowski S., Siwek K.: Selforganizing neural networks for short term load forecasting in power system. In: *Engineering Applications of Neural Networks*. Gibraltar, (1998) 253-256
17. Osowski S., Siwek K.: Regularization of neural networks for improved load forecasting in the power system. *IEE Proc. Generation, Transmission and Distribution*, **149** (2002) 340-344
18. Osowski S., Siwek K., Tran Hoai L.,: Short term load forecasting using neural networks. *Proc. III Ukrainian-Polish Workshop*. Alushta, Krym (2001) 72-77

19. Park D.C., El-Sharkawi M.A., Marks R.J., Atlas R.E., Damborg M.J.: Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems* **6** (1991) 442-449
20. Peng T.M., Hubele N.F., Karady G.G.: Advancement in the application of neural networks for short-term load forecasting. *IEEE Transactions on Power Systems* **7** (1992) 250-257
21. Siwek K.: Load forecasting in an electrical power system using artificial neural networks. PhD Thesis, Faculty of Electricity, Warsaw Technical University (2001) (in Polish).
22. Tresp V.: Committee Machines. In: Yu Hen Hu, Jenq-Neng Hwang (eds.): *Handbook for Neural Network Signal Processing*. CRC Press (2001)
23. Weron A., Weron R.: *Stock Market of Energy*. CIRE, Wrocław (2000) (in Polish)
24. Zieliński J.S.: *Intelligent Systems in Management - Theory and Practice*. PWN, Warszawa (2000) (in Polish)

# Jet Engine Turbine and Compressor Characteristics Approximation by Means of Artificial Neural Networks

Maciej Lawryńczuk

Institute of Control and Computation Engineering,  
Warsaw University of Technology,  
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland  
Tel.: +48 22 234-73-97  
M.Lawrynczuk@ia.pw.edu.pl

**Abstract.** This paper is concerned with the approximation problem of the SO-3 jet engine turbine and compressor characteristics. Topology selection of multilayer feedforward artificial neural networks is investigated. Neural models are compared with Takagi-Sugeno fuzzy models in terms of approximation accuracy and complexity.

## 1 Introduction

Advanced jet engines are very expensive, their cost exceeds million of dollars whereas experimental, prototypical engines are even more expensive. Moreover, the specialised laboratories in which real engines can undergo all the requisite tests also need significant financial expenses. That is why computer simulation methods are used in jet engines design and development as frequently as possible [1]. It should be emphasised, however, that computer simulations hinge on the quality of the model. If the model is accurate enough the results obtained off-line and the conclusions drawn during the simulations can be applied in practice to the prototypical engine. It means that the time spent on real engine tests can be reasonably reduced, which, in turn, leads to significant money savings. On the other hand, inaccurate models used in simulations are likely to result in useless designs, time and money losses.

From control engineering perspective, dynamic models of the jet engine can be employed to

- a) jet engine control system design and evaluation [1],
- b) jet engine observer system design and evaluation (virtual measurements) [9], [10],
- c) jet engine fault detection and isolation system design and evaluation.

Considering the accuracy of the whole jet engine dynamic model, the approximation precision of the turbine and compressor static characteristics is of pivotal importance [1]. These characteristics are nonlinear, in the literature the application of different approximation methods has been reported [8]. In this paper



multilayer feedforward neural networks [2], [5] and Takagi-Sugeno fuzzy models [12], [13] are used for approximation purposes. Since both model structures are proven to serve as universal approximators [6], [13], it is interesting to study their practical application. In light of using the obtained turbine and compressor approximations in the nonlinear dynamic model of the jet engine to be finally used in aforementioned system designs, not only model accuracy, but also complexity (i.e. the number of parameters) is important. For model identification and validation real data sets obtained from the SO-3 engine are used.

## 2 Turbine and Compressor Characteristics

So as to describe the properties of the jet engine turbine and compressor it is necessary to use two characteristics for each unit. They are nonlinear functions of two variables.

### 2.1 Turbine Characteristics

The first turbine characteristic is

$$\eta = f_{t\eta}(\varepsilon, n_r) \quad (1)$$

where  $\eta$  is the isentropic efficiency of the turbine,  $\varepsilon$  is the pressure ratio of the working medium flowing through the turbine and  $n_r$  [ $rpm/K^{0.5}$ ] is the reduced rotational speed of the turbine rotor.

The second turbine characteristic is

$$m_t = f_{tm}(\varepsilon, n_r) \quad (2)$$

where  $m_t$  [ $(kg \cdot K^{0.5})/(s \cdot Mpa)$ ] is the reduced mass flow of the working medium flowing through the turbine. The turbine characteristics are depicted in Fig. 1 and 2.

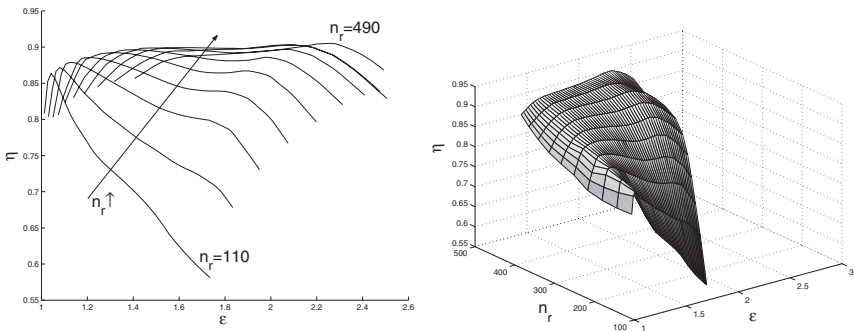
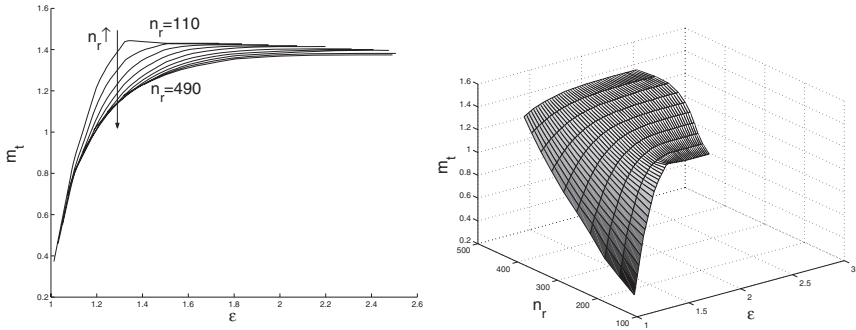


Fig. 1. 2D and 3D views of the turbine characteristic  $\eta = f_{t\eta}(\varepsilon, n_r)$



**Fig. 2.** 2D and 3D views of the turbine characteristic  $m_t = f_{tm}(\varepsilon, n_r)$

### 2.2 Compressor Characteristics

The first compressor characteristic is

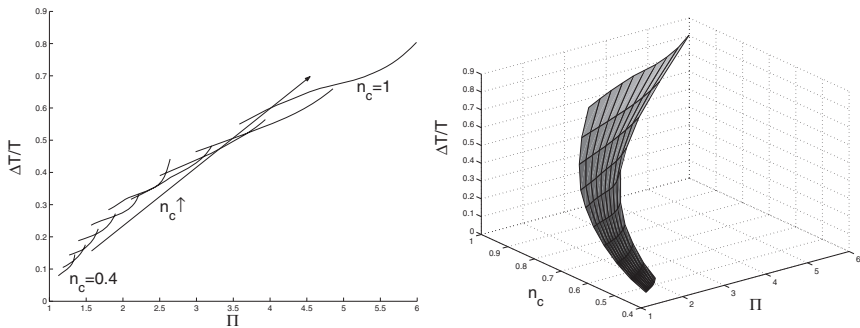
$$\frac{\Delta T}{T} = f_{ct}(\Pi, n_c) \tag{3}$$

where  $\frac{\Delta T}{T}$  is the relative increment of the total temperature of the air flowing through the compressor,  $\Pi$  is the pressure ratio of the air flowing through the compressor and  $n_c$  [rpm/15600] is the reduced rotational speed of the compressor rotor.

The second compressor characteristic is

$$m_c = f_{cm}(\Pi, n_c) \tag{4}$$

where  $m_c$  [kg/s] is the reduced mass flow of the air flowing through the compressor. The compressor characteristics are depicted in Fig. 3 and 4.



**Fig. 3.** 2D and 3D views of the compressor characteristic  $\frac{\Delta T}{T} = f_{ct}(\Pi, n_c)$

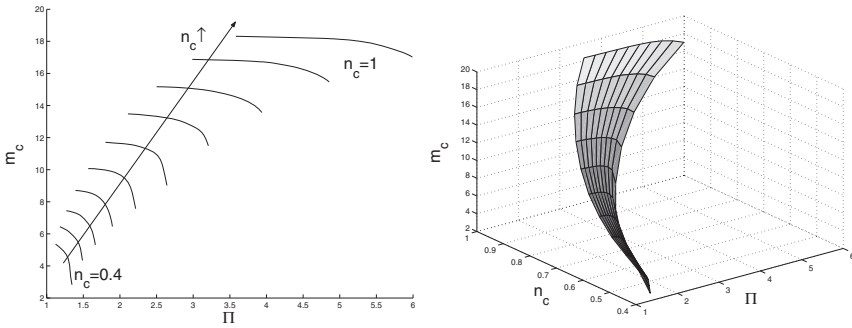


Fig. 4. 2D and 3D views of the compressor characteristic  $m_c = f_{cm}(\Pi, n_c)$

### 3 Neural and Fuzzy Approximation of Turbine and Compressor Characteristics

For approximation feedforward neural networks of multilayer structure [2], [5] and Takagi-Sugeno fuzzy models [12], [13] are used. All the models studied have two inputs and one output. Two kinds of feedforward neural networks are considered: with one hidden layer containing  $K$  hidden nodes and with two hidden layers containing  $K_1$  and  $K_2$  hidden units in the first and the second hidden layer, respectively. The hyperbolic tangent is used as nonlinear transfer function, the output is linear.

Although it is a well known and proved [6] fact that the feedforward neural network with as few as one nonlinear hidden layer can serve as the universal approximators, it is also interesting to experiment with networks containing two hidden layers. Practical experience based on huge number of simulations carried out indicates that neural structures with two hidden layers need reasonably smaller number of training epochs, which means, that the complexity and time of model identification as well as validation stages are significantly reduced.

Normalisation is very important, it affects significantly models accuracy. All input variables are normalised to the range  $[0, 1]$ , the output variables are not scaled. At first, real data sets from the SO-3 engine are obtained and verified. These sets are next used to plot the nonlinear turbine and compressor characteristics shown in Fig. 1, 2, 3 and 4. The objective is to find the nonlinear functions given in very general form by the equations (1), (2), (3), (4) which approximate the data sets precisely. In fact, for each characteristic considered, two data sets are collected. The first one, named 'the training set' (TS) is used for training purposes only to find the parameters of the model. Analogously, 'the validation set' (VS) is exclusively used to assess the model accuracy. Both sets, in the case of each characteristic, have 1500 patterns. Neural models are trained so as to minimise the Sum of Squared Errors (SSE.) SSE is also calculated for validation set to study generalisation abilities of the models.

As far as feedforward neural models are concerned, different training algorithms have been tested: the rudimentary backpropagation scheme (i.e. the steepest descent), the conjugate gradient methods (Polak-Ribiere, Fletcher-Reeves),

**Table 1.** Maximum number of training epochs  $n_{max}$  for different neural models

one hidden layer		two hidden layers	
$K$	$n_{max}$	$K_1 = K_2$	$n_{max}$
5, ..., 8	1000	6, 7	2000
9, ..., 20	2000	8	2500
21, ..., 25	2500	9, 10	3000

the quasi-Newton algorithms (DFP, BFGS) and the Levenberg-Marquardt algorithm [1], [3]. Finally, all neural models are trained using the Levenberg-Marquardt algorithm, which outperforms all the aforementioned competitors in terms of learning time. Such an observation is not surprising, since neural network training task is in fact an unconstrained minimisation problem with the SSE performance index as the objective function. Of course, the maximum number of training epochs depends on the size of the neural model. Maximum numbers of epochs for neural networks of different structure, which are determined by a trial-and-error procedure, are given in Table 1. For Takagi-Sugeno fuzzy models the standard learning algorithm is used [13].

For each neural model structure the identification experiment is repeated 10 times, the weights of the neural networks are initialised randomly. The results presented are the best obtained. Table 2 and 3 compare fuzzy and neural turbine

**Table 2.** Comparison of turbine models

		model $\eta = f_{t\eta}(\varepsilon, n_r)$		model $m_t = f_{tm}(\varepsilon, n_r)$		
	$n_{mf}$	$n_{par}$	SSE TS	SSE VS	SSE TS	SSE VS
Takagi-Sugeno fuzzy models	2	28	$1.0310 \cdot 10^{-1}$	$1.0892 \cdot 10^{-1}$	$1.0271 \cdot 10^0$	$1.1043 \cdot 10^0$
	4	80	$3.1244 \cdot 10^{-2}$	$3.6131 \cdot 10^{-2}$	$3.7222 \cdot 10^{-2}$	$3.7175 \cdot 10^{-2}$
	6	156	$3.4739 \cdot 10^{-3}$	$3.8239 \cdot 10^{-3}$	$1.2799 \cdot 10^{-2}$	$1.2216 \cdot 10^{-2}$
	8	256	$1.6026 \cdot 10^{-3}$	$3.1503 \cdot 10^{-3}$	$4.7011 \cdot 10^{-3}$	$4.7588 \cdot 10^{-3}$
	10	380	$8.0389 \cdot 10^{-4}$	$1.5080 \cdot 10^{-3}$	$2.7178 \cdot 10^{-3}$	$2.7087 \cdot 10^{-3}$
	$K$	$n_{par}$	SSE TS	SSE VS	SSE TS	SSE VS
neural models	5	21	$4.5861 \cdot 10^{-2}$	$4.8957 \cdot 10^{-2}$	$3.3766 \cdot 10^{-2}$	$3.3841 \cdot 10^{-2}$
	10	41	$8.8987 \cdot 10^{-3}$	$9.3675 \cdot 10^{-3}$	$8.2354 \cdot 10^{-3}$	$8.0531 \cdot 10^{-3}$
with one hidden layer	15	61	$4.1639 \cdot 10^{-3}$	$4.4554 \cdot 10^{-3}$	$4.8479 \cdot 10^{-3}$	$4.6290 \cdot 10^{-3}$
	20	81	$2.6460 \cdot 10^{-3}$	$2.9389 \cdot 10^{-3}$	$2.8258 \cdot 10^{-3}$	$2.7655 \cdot 10^{-3}$
	25	101	$1.5869 \cdot 10^{-3}$	$1.7260 \cdot 10^{-3}$	$2.3349 \cdot 10^{-3}$	$2.2760 \cdot 10^{-3}$
	$K_1 = K_2$	$n_{par}$	SSE TS	SSE VS	SSE TS	SSE VS
neural models	6	67	$2.0315 \cdot 10^{-3}$	$2.2353 \cdot 10^{-3}$	$2.6239 \cdot 10^{-3}$	$2.5720 \cdot 10^{-3}$
	7	85	$1.5038 \cdot 10^{-3}$	$1.7099 \cdot 10^{-3}$	$1.9590 \cdot 10^{-3}$	$1.9049 \cdot 10^{-3}$
with two hidden layers	8	105	$9.5021 \cdot 10^{-4}$	$1.0869 \cdot 10^{-3}$	$1.3774 \cdot 10^{-3}$	$1.3948 \cdot 10^{-3}$
	9	127	$5.7671 \cdot 10^{-4}$	$6.7594 \cdot 10^{-4}$	$9.9892 \cdot 10^{-4}$	$1.0275 \cdot 10^{-3}$
	10	151	$4.1110 \cdot 10^{-4}$	$4.8272 \cdot 10^{-4}$	$8.6042 \cdot 10^{-4}$	$8.6660 \cdot 10^{-4}$

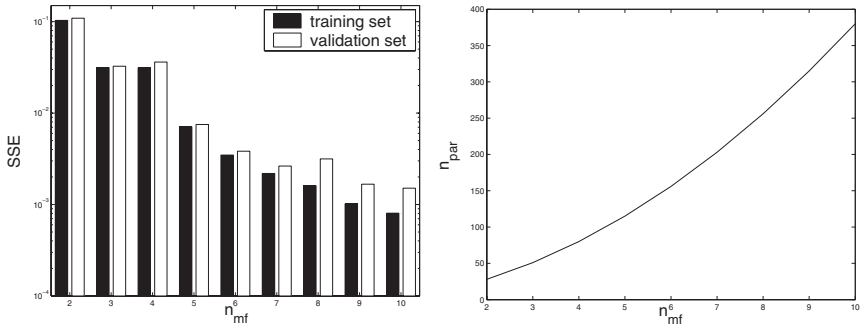
**Table 3.** Comparison of compressor models

		model $\frac{\Delta T}{T} = f_{ct}(II, n_c)$		model $m_c = f_{cm}(II, n_c)$			
		$n_{mf}$	$n_{par}$	SSE TS	SSE VS	SSE TS	SSE VS
Takagi-Sugeno fuzzy models		2	28	$1.1788 \cdot 10^{-1}$	$1.1816 \cdot 10^{-1}$	$4.5684 \cdot 10^1$	$4.7337 \cdot 10^1$
		4	80	$9.6945 \cdot 10^{-3}$	$9.7923 \cdot 10^{-3}$	$1.2489 \cdot 10^1$	$1.3502 \cdot 10^1$
		6	156	$4.4854 \cdot 10^{-3}$	$4.7330 \cdot 10^{-3}$	$7.2516 \cdot 10^0$	$7.8373 \cdot 10^0$
		8	256	$1.8051 \cdot 10^{-3}$	$2.9723 \cdot 10^{-3}$	$3.8626 \cdot 10^0$	$4.7926 \cdot 10^0$
		10	380	$1.0713 \cdot 10^{-3}$	$2.2536 \cdot 10^{-3}$	$2.6320 \cdot 10^0$	$1.2154 \cdot 10^0$
		$K$	$n_{par}$	SSE TS	SSE VS	SSE TS	SSE VS
neural models with one hidden layer		5	21	$4.5378 \cdot 10^{-2}$	$4.6749 \cdot 10^{-2}$	$3.3204 \cdot 10^1$	$3.6554 \cdot 10^1$
		10	41	$1.2011 \cdot 10^{-2}$	$1.2372 \cdot 10^{-2}$	$1.1626 \cdot 10^1$	$1.2927 \cdot 10^1$
		15	61	$6.7304 \cdot 10^{-3}$	$6.8970 \cdot 10^{-3}$	$4.8644 \cdot 10^0$	$5.5854 \cdot 10^0$
		20	81	$2.9716 \cdot 10^{-3}$	$3.0621 \cdot 10^{-3}$	$2.5734 \cdot 10^0$	$2.9479 \cdot 10^0$
		25	101	$1.5074 \cdot 10^{-3}$	$1.8046 \cdot 10^{-3}$	$2.0165 \cdot 10^0$	$2.4870 \cdot 10^0$
		$K_1 = K_2$	$n_{par}$	SSE TS	SSE VS	SSE TS	SSE VS
neural models with two hidden layers		6	67	$1.1421 \cdot 10^{-3}$	$1.5320 \cdot 10^{-3}$	$8.6382 \cdot 10^{-1}$	$1.2224 \cdot 10^0$
		7	85	$8.0815 \cdot 10^{-4}$	$1.1555 \cdot 10^{-3}$	$2.6174 \cdot 10^{-1}$	$3.9231 \cdot 10^{-1}$
		8	105	$5.9702 \cdot 10^{-4}$	$8.2032 \cdot 10^{-4}$	$1.9111 \cdot 10^{-1}$	$3.1918 \cdot 10^{-1}$
		9	127	$2.3266 \cdot 10^{-4}$	$5.7551 \cdot 10^{-4}$	$1.2084 \cdot 10^{-1}$	$2.3151 \cdot 10^{-1}$
		10	151	$1.1234 \cdot 10^{-4}$	$3.2241 \cdot 10^{-4}$	$5.5645 \cdot 10^{-2}$	$1.9866 \cdot 10^{-1}$

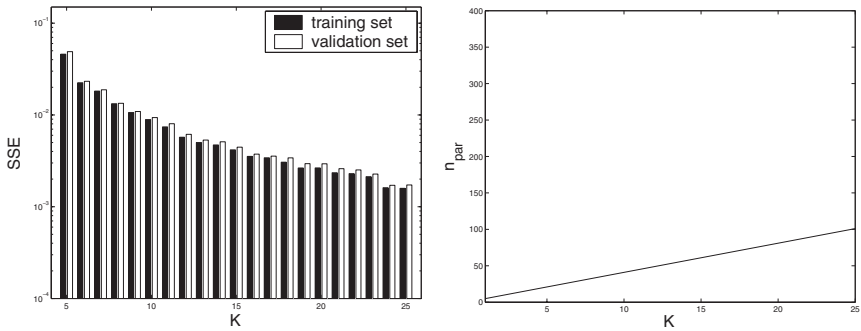
and compressor models in terms of SSE calculated for both training and validation sets and the number of parameters  $n_{par}$ .

For instance, let us study and discuss the approximation of the first turbine characteristic  $\eta = f_{t\eta}(\varepsilon, n_r)$ . The influence of the number of membership functions  $n_{mf}$  (equal for both inputs) of fuzzy turbine model on SSE and on the number of parameters is depicted in Fig. 5. As the approximation accuracy increases, the number of parameters soars ("the curse of dimensionality" phenomenon.) The influence of the number of hidden nodes of neural turbine model with one and two hidden layers on SSE and on the number of parameters is shown in Fig 6 and 7. When compared with the fuzzy models, both neural models give better approximation accuracy, whereas their number of parameters is moderate. Fig. 8 compares the turbine neural model with one hidden layer containing only  $K = 5$  nodes and real data. Of course, such a neural network has too few parameters, it is unable to approximate precisely enough the nonlinear characteristic. The best obtained result for the turbine neural model with two hidden layers containing  $K_1 = K_2 = 10$  nodes vs. real data is shown in Fig. 9.

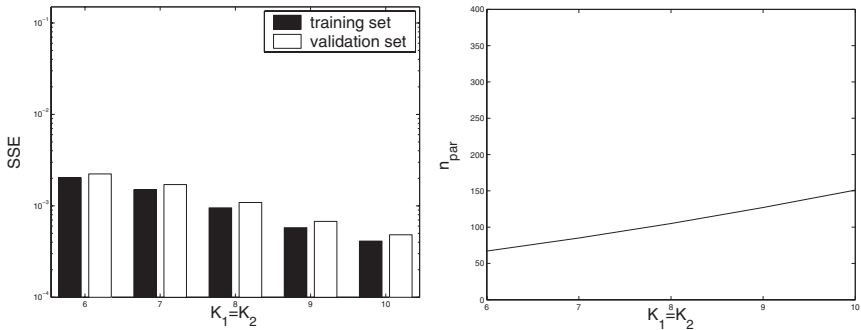
Comparing fuzzy models, neural models with one and two hidden layers, it can be concluded that the last structure, for given maximum numbers of training epochs (Table II), turns out to learn fastest and is most accurate. Fuzzy models are less accurate and have vast number of parameters. For example, in case of



**Fig. 5.** The influence of the number of membership functions  $n_{mf}$  of fuzzy turbine model  $\eta = f_{t\eta}(\varepsilon, n_r)$  on SSE (*left*) and on the number of parameters  $n_{par}$  (*right*)

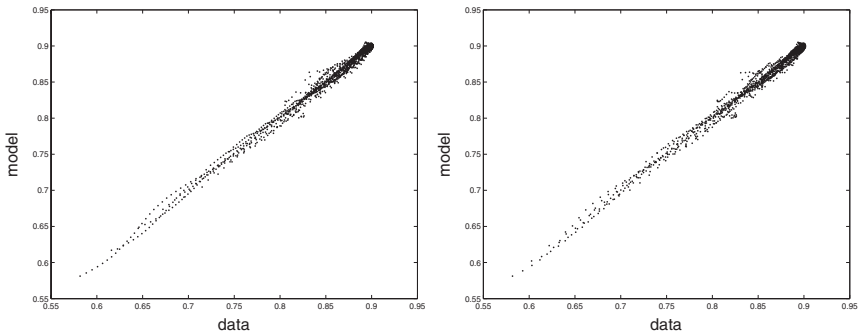


**Fig. 6.** The influence of the number of hidden nodes of neural turbine model  $\eta = f_{t\eta}(\varepsilon, n_r)$  with one hidden layer containing  $K$  nodes on SSE (*left*) and on the number of parameters  $n_{par}$  (*right*)

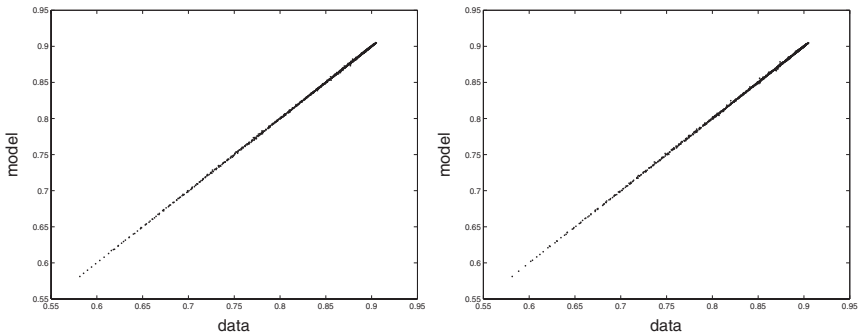


**Fig. 7.** The influence of the number of hidden nodes of neural turbine model  $\eta = f_{t\eta}(\varepsilon, n_r)$  with two hidden layers containing  $K_1 = K_2$  nodes on SSE (*left*) and on the number of parameters  $n_{par}$  (*right*)

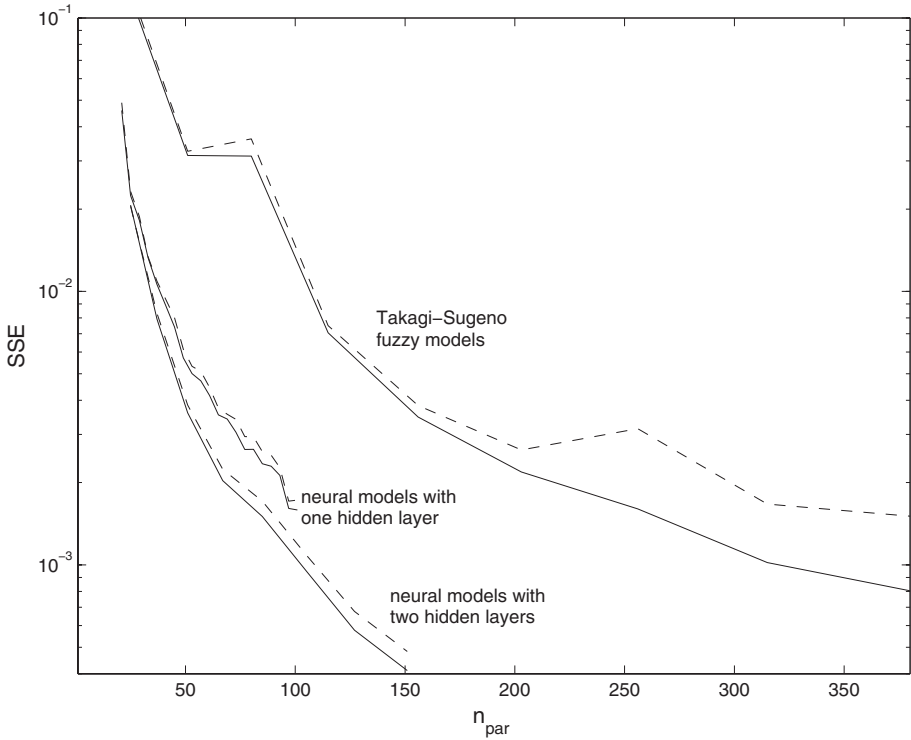
all four models considered, double-layered networks with  $K_1 = K_2 = 7$  nodes ( $n_{par} = 85$ ) outperform single-layered ones with  $K = 25$  nodes ( $n_{par} = 101$ ) and all fuzzy models with  $n_{mf} = 8$  ( $n_{par} = 256$ .) Apart from the first turbine characteristic, the double-layered neural models with  $K_1 = K_2 = 7$  is much more accurate than the fuzzy models with  $n_{mf} = 10$  ( $n_{par} = 380$ .) Fig. 10 compares the influence of the number of model parameters of the turbine neural and fuzzy models on SEE. In general, fuzzy models are not as accurate as neural ones and have huge number of parameters. Moreover, for complicated fuzzy models (containing many parameters) the generalisation abilities are poor (big SEE for the validation set.) In order to further reduce the number of weights the neural network can be pruned, using, for instance, OBD (Optimal Brain Damage) [7] or OBS (Optimal Brain Surgeon) [4] algorithms.



**Fig. 8.** Turbine neural model  $\eta = f_{t\eta}(\varepsilon, n_r)$  with one hidden layer containing  $K = 5$  nodes vs. real data: the training set (*left*) and the validation set (*right*)



**Fig. 9.** Turbine neural model  $\eta = f_{t\eta}(\varepsilon, n_r)$  with two hidden layers containing  $K_1 = K_2 = 10$  nodes vs. real data: the training set (*left*) and the validation set (*right*)



**Fig. 10.** The influence of the number of parameters  $n_{par}$  of turbine neural and fuzzy models  $\eta = f_{t\eta}(\varepsilon, n_r)$  on SEE: the training set (*solid line*) and the validation set (*dashed line*)

## 4 Conclusion

This paper discusses the approximation problem of the SO-3 jet engine turbine and compressor static characteristics by means of multilayer feedforward neural networks and Takagi-Sugeno fuzzy models. The problem is particularly important since the accuracy of the whole nonlinear jet engine dynamic model is determined by the approximation precision of these nonlinear characteristics. Although both considered model structures are proven to serve as universal approximators [6], [13], the results of the experiments carried out clearly indicate that the feedforward neural networks merit serious consideration. Not only have they excellent approximation abilities but also the resulting neural models are relatively simple, when compared with fuzzy models they do not suffer from "the curse of dimensionality" phenomenon. Hence, the feedforward neural models can be easily incorporated into the nonlinear dynamic model of the jet engine and finally used in jet engine control system design, observer system design or fault detection and isolation system design.



**Acknowledgements.** The author is grateful to Dr Wojciech Izydor Pawlak from Air Force Institute of Technology, Warszawa, for thorough introduction to the field of jet engine modelling and providing data sets used in the experiments.

## References

1. M. S. Bazaraa, J. Sherali, K. Shetty: *Nonlinear programming: theory and algorithms*. Prentice Hall. (1999)
2. Bishop, C. M.: *Neural networks for pattern recognition*. Oxford University Press. Oxford. (1995)
3. Bishop, P., Murray, W., Wright, M.: *Practical optimization*. Academic Press. New York. (1981)
4. Hassibi, B., Stork, B.: Second order derivatives for network pruning: Optimal brain surgeon. *Advances of NIPS5*. Morgan Kaufmann. San Mateo. (1993) 164–171
5. Haykin, S.: *Neural networks – a comprehensive foundation*. John Wiley & Sons. New York. (1993)
6. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. *Neural networks*. **2** (1989) 359–366
7. LeCun, Y., Denker, J., Solla, S.: Optimal brain damage. *Advances of NIPS2*. Morgan Kaufmann. San Mateo. (1990) 598–605
8. Orkisz, M., Stawarz, S.: Modeling of turbine engine axial-flow compressor and turbine characteristics. *Journal of Propulsion and Power*. **16** (2000) 336–341
9. Pawlak, W. I.: Nonlinear observer of a single-flow single-flow jet turbine engine during operation. *Journal of KONES International Combustion Engines*. European Science Society of Powertrain and Transport Publication. (2005)
10. Pawlak, W. I.: Nonlinear observer in control system of a turbine jet engine. *The Archive of Mechanical Engineering*. **L** (2003) 227–245
11. Pawlak, W. I., Wiklik, K., Morawski, M. J.: Jet engine control system design and development by means of computer simulation methods (in Polish). *Scientific Library of the Institute of Aviation*. Warsaw. (1996)
12. Takagi, T, Sugeno M.: Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*. **15** (1985) 116–132.
13. Wang, L. X.: *Adaptive fuzzy systems and control. Design and stability analysis*. Prentice Hall. New Jersey. (1994)

# Speech Enhancement System Based on Auditory System and Time-Delay Neural Network

Jae-Seung Choi<sup>1</sup> and Seung-Jin Park<sup>2</sup>

<sup>1</sup> Department of Electronics Engineering, Silla University, San 1-1  
Gwaebop-dong, Sasang-gu, Busan, Korea  
choijs@knu.ac.kr

<sup>2</sup> Department of Biomedical Engineering, Chonnam National University Hospital & Medical  
School, Gwangju, Korea  
sjinpark@jnu.ac.kr

**Abstract.** This paper proposes a speech enhancement system based on an auditory system for noise reduction in speech that is degraded by background noises. Accordingly, the proposed system adjusts frame by frame the coefficients for both lateral inhibition and amplitude component according to the detected sections for each input frame, then reduces the noise signal using a time-delay neural network. Based on measuring signal-to-noise ratios, experiments confirm that the proposed system is effective for speech that is degraded by various noises.

## 1 Introduction

In recent years, the performance of speech recognition systems has improved, resulting in their adoption in many practical applications. System reliability, and in particular, the speech recognition ratio, however, can be significantly decreased by background noise [1]. Typical background noise cannot be simply eliminated with a Wiener filter [2], or spectral subtraction [3], but more skillful techniques are required. Thus, to solve this problem, this paper proposes a speech enhancement algorithm using an auditory system [4, 5, 6] and a time-delay neural network (TDNN) [7] that are effective in various noisy environments.

In speech signal processing, the major application of a neural network (NN) is the category classification of phoneme recognition, while in the area of speech enhancement and noise reduction, the major application of an NN is the extraction of speech sections [8] from a noisy speech signal. Thus, for speech signal processing, the NN needs to be constructed using a time structure, as time variation is significant information. Moreover, an amplitude component contains more information than a phase component when a speech signal is generated by a fast Fourier transform (FFT).

The purpose of the current research is the technical application of lateral inhibition rather than the imitation of a physiological auditory mechanism. Therefore, the proposed system adjusts, in a frame-by-frame manner, the two optimal parameters of the lateral inhibition filter (i.e. an adjustable parameter for lateral inhibition and a parameter of amplitude component), then the FFT amplitude component is restored using a TDNN, which includes a time structure in the NN as a method of spectrum recovery.

## 2 Experimental Conditions

The original speech signal is assumed to be  $s(t)$ , and the speech signal disturbed by additive noise is given as  $x(t) = s(t) + n(t)$ . Here,  $n(t)$  is additive noise with a sampling frequency of 8 kHz. The speech data used in the experiment were of the Aurora2 database and they consist of English connected digits recorded in clean environments with a sampling frequency of 8 kHz. The proposed system was evaluated using clean speech data from the Aurora2 database in Test Sets A, B, and C and five types of background noise, i.e. car and subway noise in Test Set A, restaurant and street noise in Test Set B, and white noise generated by a computer program.

## 3 Method for Improving Speech Characteristics

To reduce unexpected peaks that are irregular between frames, a spectral average is adopted

$$\bar{P}(i, \omega) = \frac{1}{2M + 1} \sum_{j=-M}^M W_j P(i - j, \omega)$$

In this paper,  $M = 2$ ,  $W_{-2} = W_2 = 0.7$ ,  $W_{-1} = W_1 = 1.1$ , and  $W_0 = 1.4$ . Where,  $\bar{P}(i, \omega)$  is the short-time power spectral average of the  $i$ th frame and  $P(i, \omega)$  is the original spectrum.

The current study uses the function of spectral lateral inhibition (FSLI) [4, 5, 6] in the frequency domain, thereby emphasizing the spectral peaks of speech by an excitation area, while simultaneously compressing noise spectral valleys by two inhibition areas, as shown in Fig. 1.

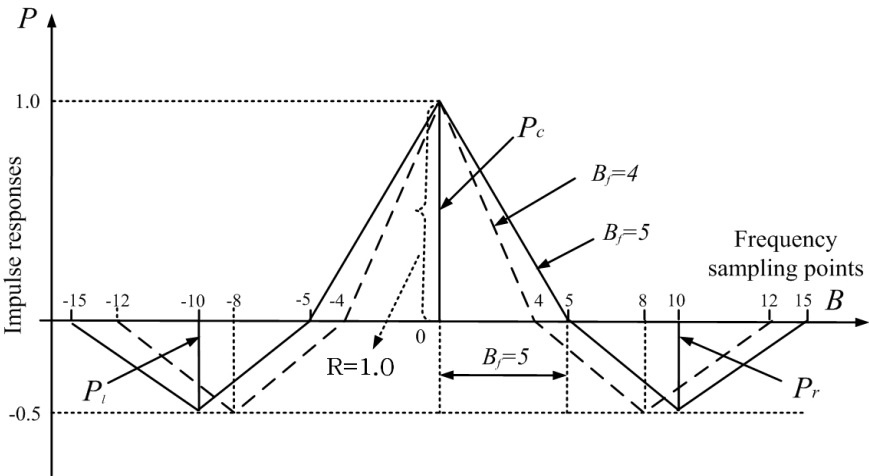


Fig. 1. Impulse responses of FSLI models

Fig. 1 shows the impulse responses for the two kinds of FSLIs selected for the voiced and unvoiced sections of each input frame, where  $B_f$  and  $R$  are the parameters that determine the width and amplitude of each FSLI, respectively. The parameters for  $P_{j=l,c,r}$  show the amplitude of the impulse response and are restricted to satisfy Equation  $P_l + P_c + P_r = 0$  for noise cancellation. In this experiment,  $P_c = 1$  and  $P_l = P_r = -0.5$ . In lateral inhibition, since the average value of the sum of the weighted noise is zero by this restriction, noise is reduced.

## 4 The Design of the Speech Enhancement System

The proposed speech enhancement system is shown in Fig. 2.

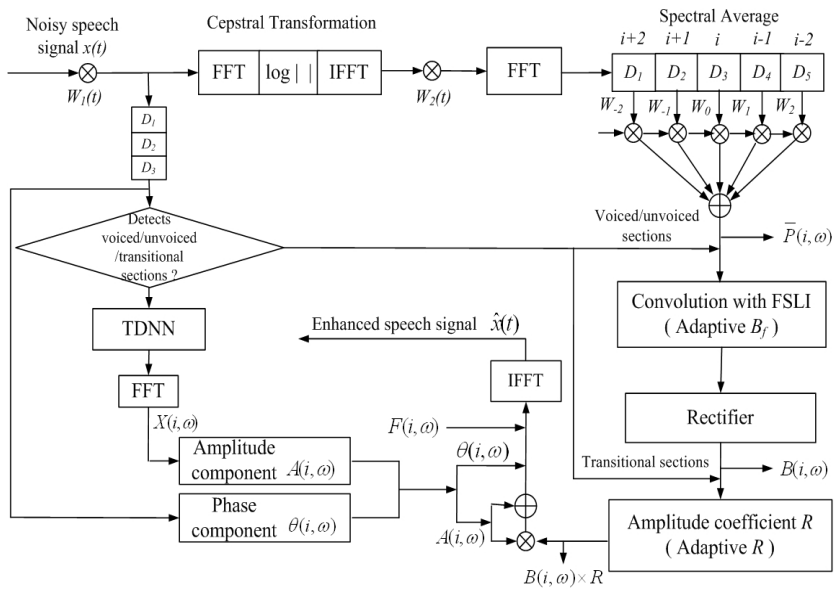


Fig. 2. The proposed speech enhancement system

First, the noisy speech signal  $x(t)$  is divided into length frames of 128 samples (16 ms), then it undergoes a cepstral transformation after passing through a Hamming window,  $W_1(t)$ . After passing through another window,  $W_2(t)$ , ten cepstrum components are obtained, from zero to the 9th in the low region, then spectral components for the speech signal are obtained by FFT. Next, noisy speech is delayed to three frames (48 ms) and is averaged based on the weighted spectral sum for each frame. Furthermore, spectral components obtained by the spectral average are convolved with the FSLI in the frequency domain. Thus, the output of the FSLI filter,  $B(i, \omega)$ , is obtained by the convolution of  $\bar{P}(i, \omega)$  with the impulse responses. Although the resulting amplitude spectrum after convolution may have some negative values, they do not have any useful information in the present situation and are set at

zero by rectification (via the above route). Meanwhile, the noisy speech signal,  $x(t)$ , in another route is delayed by three frames, and is reduced by the proposed TDNN after detecting voiced, unvoiced, or transitional sections. Then,  $B_f$  and  $R$  are adjusted to the optimal value, in a frame-by-frame manner, when the detected results for each frame are voiced and unvoiced sections. Moreover, when the detected results for each frame are transitional sections, the spectral components obtained by the spectral average are passed through to the amplitude coefficient  $R$  block without processing the FSLI and rectification. After obtaining the amplitude  $A(i, \omega)$  and phase components  $\theta(i, \omega)$ , the enhanced speech signal  $\hat{x}(t)$  is finally regenerated by IFFT using the following Equation  $F(i, \omega) = A(i, \omega)(1 + R \times B(i, \omega))e^{j\theta(i, \omega)}$ , where  $i$  and  $\omega$  represent the frame and spectral numbers, respectively.

In general, a short-term energy threshold is calculated from the start of every noise section. The maximum value for the threshold calculated at the start of five sections, however, was used in the experiments. The proposed system detects voiced sections in every frame where  $R_f \geq T_h$ , unvoiced sections in every frame where  $T_h < R_f \leq T_h/\alpha$ , and transitional sections in every frame where  $R_f < T_h/\alpha$ . Here,  $R_f$  is the effective value obtained for each frame,  $T_h$  is the threshold value aforementioned; and  $\alpha$  is a constant that is determined via experiments.

## 5 Time-Delay Neural Network (TDNN)

This paper proposes an algorithm using improved TDNNs that are separated into voiced and unvoiced sections, making the TDNN easier to train according to a somewhat similar pattern. TDNNs are also constructed for low and high-frequency bands, allowing for more effective correlation of added information. The proposed TDNN were trained using a back propagation algorithm.

The proposed TDNNs are composed of four layers and the compositions of the TDNNs are 32-64-32-32. A time series of 32-unit FFT amplitude components are fed into the input layer with  $n$  frames. Thereafter, four frames in the input layer are connected to a frame in the first hidden layer. Every six frames in the first hidden layer, with 64 units, are connected to a frame in the second hidden layer. Then, every frame in the second hidden layer, with 32 units, is connected to the output layer. In this experiment, the input signals for the TDNN, with a low-frequency band, are between zero and the 31st samples (0 kHz to 1.9 kHz) of the FFT amplitude component, where the input signals consist of the target frame, two previous frames, and the next frame. The target signals are between zero and the 31st samples of the FFT amplitude component, with a frame corresponding to a training signal for clean speech signals. Meanwhile, the input signals for the TDNN, with a high-frequency band, represent between the 32nd and 63rd samples (2 kHz to 3.9 kHz) of the FFT amplitude component, whereby the input signals also contain additional frames. The target signals are found in between the 32nd and 63rd samples of the FFT amplitude component, with a frame corresponding to a training signal for clean speech signals. In this experiment, the proposed TDNNs were trained using five kinds of network: (1) input signal-to-noise ratio ( $SNR_{in}$ ) = 20 dB, (2)  $SNR_{in}$  = 15 dB, (3)  $SNR_{in}$  = 10 dB,

(4)  $SNR_{in} = 5$  dB, and (5)  $SNR_{in} = 0$  dB. Thus, a total of thirty simulations were used to train the NN, for one network. When using the Aurora2 database, the TDNNs were trained after adding white, car, restaurant, and subway noise to the clean speech data in Test Set A. In training, the training coefficient was set to 0.2 and the inertia coefficient was set to 0.6. When the training iterations exceeded 10,000, there was almost no decrease in training error curves at minimum error points. Therefore, 10,000 was set as the maximum number of training iterations for the experiment.

Fig. 3 shows how the TDNN input signals are utilized after training. First, noisy speech signals  $x(t)$  are detected in voiced and unvoiced sections, then they are separated into FFT amplitude components according to the voiced and unvoiced sections. Thereafter, the separated FFT amplitude components are added to the appropriate TDNN input signals with either a low or high-frequency band. The final FFT amplitude components are obtained by combining the results from the TDNNs with low and high-frequency bands, however, the phase components come directly from the FFT. Thereafter, enhanced speech signals  $y(t)$  are regenerated using the IFFT.

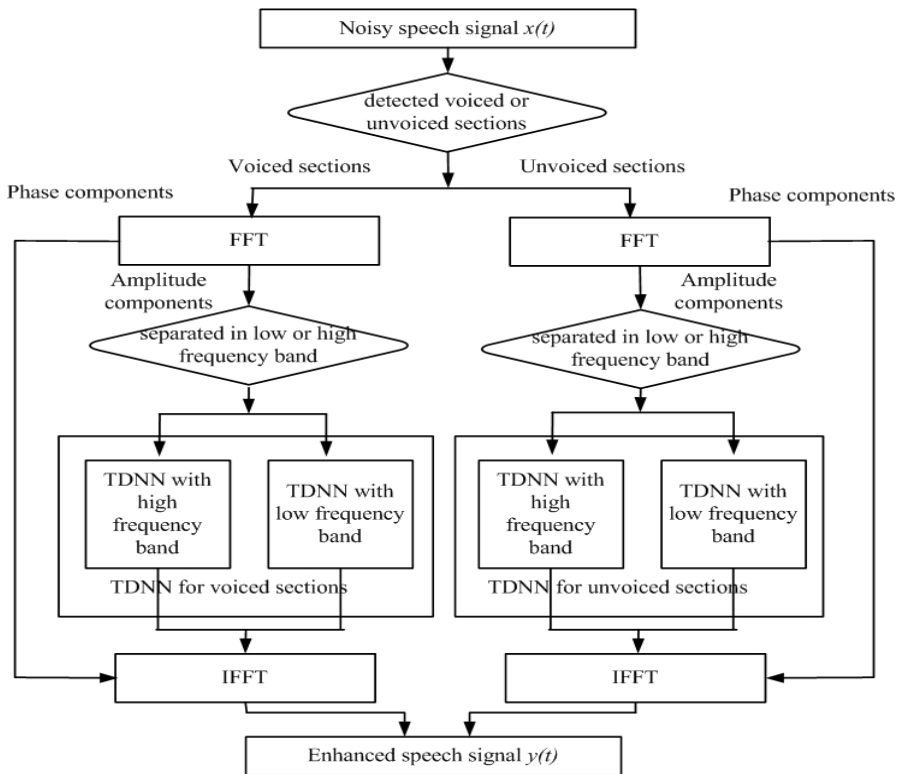


Fig. 3. The proposed TDNNs after training

## 6 Experimental Results and Considerations

Using the basic composition conditions described above, experiments confirmed that the proposed system was effective for speech degraded by additive white, car, restaurant, subway, and street noise based on measuring the SNR.

### 6.1 The Effects of Coefficients $B_f$ and $R$

To obtain optimal values for the parameters of  $B_f$  and  $R$ , the SNR was used. For comparison, twenty sentences were randomly selected from the Aurora2 database in Test Set A. Fig. 4 shows the averages of the approximate output SNR ( $SNR_{out}$ ) values obtained when adjusting  $R$  for each  $B_f$  in the case of voiced and unvoiced sections of car noise. From the  $SNR_{out}$  evaluation values listed in Fig. 4, the optimal values of  $R$  for each  $SNR_{in}$  were obtained by adjusting  $R$ . For instance, in the case of  $SNR_{in} = 0\text{dB}$  in the voiced sections of Fig. 4(a), the maximum  $SNR_{out}$  value was approximately 12.5 dB, so the optimal value for  $R$  was 2.0. Therefore, the  $SNR_{out}$  value was improved by approximately 12.5 dB as compared to the  $SNR_{in}$  value of 0 dB for the original noisy speech when  $R = 0.0$ . Here,  $R = 0.0$  represents the baseline value of  $SNR_{out} = SNR_{in}$  for the original speech signal degraded by car noise. Although not shown in the Figure, the only values used for  $B_f$  were 4 and 5, as no other values were effective for the  $SNR_{out}$  in this experiment. When using the proposed speech enhancement system, the parameters were adjusted to  $B_f = 5$  and  $R = 2.0$  when the detected results for each frame were voiced sections, and  $B_f = 4$  and  $R = 1.0$  when the detected results were unvoiced sections.

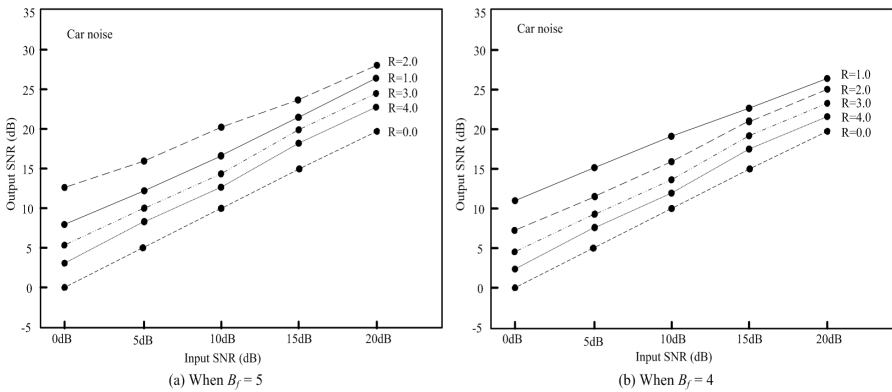


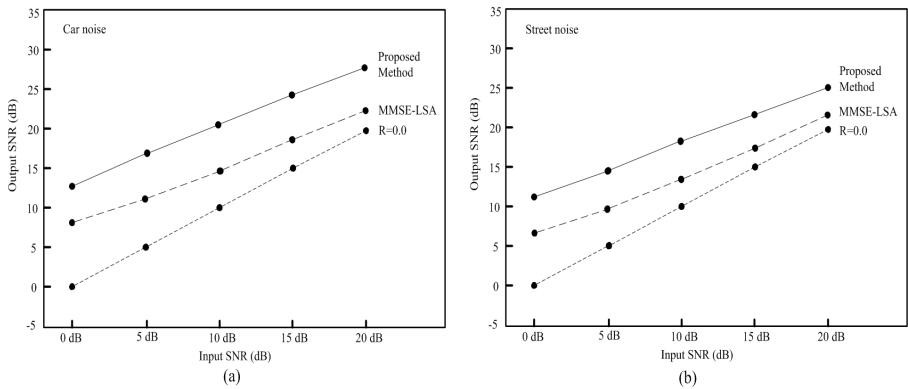
Fig. 4. The effect of  $B_f$  and  $R$ ; (a) voiced sections, (b) unvoiced sections

### 6.2 Performance Evaluation and Comparison for Speech Enhancement

To evaluate performance, noisy speech data were randomly selected from the Aurora2 database of Test Sets B and C. For comparison, the proposed system was compared with a minimum mean-square error log-spectral amplitude (MMSE-LSA) [9] estimator for white, car, restaurant, subway, and street noise. The MMSE-LSA

estimator is based on a minimum mean-square error short-time spectral amplitude (MMSE-STSA) estimator [10], which can be derived by modeling speech and noise spectral components as statistically independent Gaussian random variables. An important property of the MMSE-LSA estimator is that it can reduce “musical noise” in enhanced speech signals. When implementing the MMSE-LSA method, the frame length was 128 samples (16 ms) and the overlap was 64 samples (8 ms). At each frame, a Hamming window was used.

Fig. 5 shows the averages of the approximate  $SNR_{out}$  over thirty sentences for the proposed system, as compared with the MMSE-LSA method, at different noise levels ( $SNR_{in} = 20 \text{ dB} \sim 0 \text{ dB}$ ) for each noise. In the case of stationary noise, as shown in Fig. 5(a), the maximal improvement in  $SNR_{out}$  values with the proposed method was approximately 6 dB better for car noise, when compared with that of the MMSE-LSA method. Moreover, a similar tendency was found for non-stationary noise, as shown in Fig. 5(b), i.e. the maximal improvement in  $SNR_{out}$  values, with the proposed method, was approximately 4.5 dB better for street noise, when compared with that of the MMSE-LSA method. Although not shown in the Figure, the  $SNR_{out}$  values for restaurant and subway noise showed a similar improvement to that of car noise when compared with that of the MMSE-LSA method. In addition, the proposed system was much better with higher noise levels than lower noise levels as shown in the Figures. The maximal improvement in the  $SNR_{out}$  values with the proposed method, was approximately 12.5 dB better for car noise and 11 dB better for street noise, when compared with  $R = 0.0$ . Thus, this 12.5 dB improvement was quite significant and evident when listening to the output. Therefore, according to the experimental results, the proposed speech enhancement system based on the FSLI and TDNN was effective for various noises.



**Fig. 5.** A comparison of the proposed and MMSE-LSA methods; (a) addition of car noise, (b) addition of street noise

## 7 Conclusions

A speech enhancement system was proposed that uses a lateral inhibition mechanism model and TDNN to reduce background noise. The experimental results were as follows:



1. There is an optimal  $B_f$  and  $R$  for each  $SNR_{in}$ , resulting in good quality speech, especially with a higher value.
2. The possibility of noise reduction using TDNNs was confirmed without depending on the content of the training set.
3. The SNR of noisy speech was improved when using the optimal  $B_f$  and  $R$ , which are the optimal estimated values per frame in the proposed system.
4. Noise reduction was significant under  $SNR_{in}$  conditions of up to 0 dB for sentences.

As mentioned above, experiments confirmed that the proposed system was effective for white, car, restaurant, subway, and street noise, as demonstrated by the improved SNR values. Therefore, it is believed that the present research will be useful in enhancing speech under noisy conditions.

## References

1. J. T. Chien, L. M. Lee, H. C. Wang, Noisy speech recognition by using variance adapted hidden Markov models, IEE Electronics Letters, Vol. 31, No. 18, (1995) pp. 1555-1556.
2. T. V. Sreenivas, P. Kirnapure, Codebook constrained wiener filtering for speech enhancement, IEEE Transactions on Speech and Audio Processing, Vol. 4, No. 5, (1996) pp. 383-389.
3. S. F. Boll, Suppression of acoustic noise in speech using spectral subtraction, IEEE Transactions on Acoustics, Speech, Signal Processing, Vol. 27, No. 2, (1979) pp. 113-120.
4. S. A. Shamma, Speech Processing in the Auditory System II: Lateral Inhibition and the Central Processing of Speech Evoked Activity in the Auditory Nerve, The Journal of the Acoustical Society of America, Vol. 78, No. 7, (1985) pp. 1622-1632.
5. Y. M. Cheng, D. O'Shaughnessy, Speech enhancement based conceptually on auditory evidence, IEEE Trans. Signal Processing. Vol. 39, No. 9, (1991) pp. 1943-1954.
6. J. H. L. Hansen, S. Nandkumar, Robust Estimation of Speech in Noisy Backgrounds Based on Aspects of the Auditory Process, The Journal of the Acoustical Society of America, Vol. 97, No. 6, (1995) pp. 3833-3849.
7. A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, Phoneme Recognition using Time-delay Neural Networks, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 37, No. 3, (1989) pp. 328-339.
8. Y. Wu and Y. Li, Robust speech/non-speech detection in adverse conditions using the fuzzy polarity correlation method, IEEE International Conference on Systems, Man, and Cybernetics, Vol. 4, (2000) pp. 2935-2939.
9. Y. Ephraim and D. Malah, "Speech Enhancement Using a Minimum Mean-Square Error Log-Spectral Amplitude Estimator", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 33, No. 2, (1985) pp. 443-445.
10. Y. Ephraim, D. Malah, Speech Enhancement Using a Minimum-Mean Square Error Short-Time Spectral Amplitude Estimator, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 32, No. 6, (1984) pp. 1109-1121.

# Recognition of Patterns Without Feature Extraction by GRNN

Övünç Polat and Tülay Yıldırım

Electronics and Communications Engineering Department  
Yıldız Technical University  
Besiktas, Istanbul 34349, Turkey  
{opolat, tulay}@yildiz.edu.tr

**Abstract.** Automatic pattern recognition is a very important task in many applications such as image segmentation, object detection, etc. This work aims to find a new approach to automatically recognize patterns such as 3D objects and handwritten digits based on a database using General Regression Neural Networks (GRNN). The designed system can be used for both 3D object recognition from 2D poses of the object and handwritten digit recognition applications. The system does not require any preprocessing and feature extraction stage before the recognition. Simulation results show that pattern recognition by GRNN improves the recognition rate considerably in comparison to other neural network structures and has shown better recognition rates and much faster training times than that of Radial Basis Function and Multilayer Perceptron networks for the same applications.

## 1 Introduction

Pattern classification is a core task in many applications such as image segmentation, object detection, etc. Many classification approaches are feature-based, which means some features have to be extracted before classification can be carried out. Explicit feature extraction is not easy, and not always reliable in some applications [1].

In this paper, our purpose is to classify patterns without feature extraction using General Regression Neural Network (GRNN).

The General Regression Neural Network which is a kind of radial basis networks was developed by Specht [2] and is a powerful regression tool with a dynamic network structure. The network training speed is extremely fast. Due to the simplicity of the network structure and its implementation, it has been widely applied to a variety of fields including image processing. Specht [3] addressed the basic concept of inclusion of clustering techniques in the GRNN model.

A pattern recognition system generally consists of three blocks. The first block is a preprocessing stage, the second block is used for feature extraction and the last block is the classification stage. The studies present in literature that uses the same object database as ours typically uses either principal component analysis (PCA) [4] or the optical flow of the image evaluated from its rotational poses [5] or by choosing the

components of different directions with Cellular Neural Network and selecting the remaining pixel numbers in that particular direction as features [6] for feature extraction.

There are convolutional neural networks which do not have a feature extraction stage, where basically the input image itself is applied at the input of the network for applications such as face recognition and handwritten character recognition in literature. In those systems, the input images are convolved with certain masks to get certain feature maps, then a subsampling stage takes place to reduce the image size and multilayer perceptron (MLP) networks are used for classification in the last stage of the network. This network topology has been applied in particular to image classification when sophisticated preprocessing is to be avoided [7], [8], [9].

In this study, we do not have a feature extraction stage. The input images are resized before they are applied to the input of the GRNN network for object recognition. The proposed system is tested for both object recognition and handwritten digit recognition applications. The system has high recognition ratio for especially object recognition despite very few reference data are used.

In this work, object recognition is achieved by comparing unknown object images with the reference set. To this end, a 128x128 size Red-Green-Blue (RGB) input image is converted to grayscale and resized as 16x16. Then the 16x16 matrix is converted to a 1x256 input vector. The input images are resized for a faster simulation. Some of poses are chosen as reference set and then the GRNN is simulated. The rest of the poses are used for test and high test performance is obtained.

As stated before, the classification is done using the proposed system without any feature extraction and preprocessing stages for the digits in the data set in the handwritten digit recognition application.

## 2 Overview of GRNN Structure

The GRNN predicts the value of one or more dependent variables, given the value of one or more independent variables. The GRNN thus takes as an input vector  $x$  of length  $n$  and generates an output vector (or scalar)  $y'$  of length  $m$ , where  $y'$  is the prediction of the actual output  $y$ . The GRNN does this by comparing a new input pattern  $x$  with a set of  $p$  stored patterns  $x^i$  (pattern nodes) for which the actual output  $y_i$  is known. The predicted output  $y'$  is the weighted average of all these associated stored outputs  $y_{ij}$ . Equation(1) expresses how each predicted output component  $y'_j$  is a function of the corresponding output components  $y_j$  associated with each stored pattern  $x^i$ . The weight  $W(x, x^i)$  reflects the contribution of each known output  $y_i$  to the predicted output. It is a measure of the similarity of each pattern node with the input pattern [10].

$$y'_j = \frac{N_j}{D} = \frac{\sum_{i=1}^p y_{ij} W(x, x^i)}{\sum_{i=1}^p W(x, x^i)}, \quad j = 1, 2, \dots, m. \quad (1)$$

It is clear from (1) that the predicted output magnitude will always lie between the minimum and maximum magnitude of the desired outputs ( $y_{ij}$ ) associated with the stored patterns (since  $0 \leq W \leq 1$ ). The GRNN is best seen as an interpolator, which interpolates between the desired outputs of pattern layer nodes that are located near the input vector (or scalar) in the input space [10].

A standard way to define the similarity function  $W$ , is to base it on a distance function,  $D(x_1, x_2)$ , that gives a measure of the distance or dissimilarity between two patterns  $x_1$  and  $x_2$ . The desired property of the weight function  $W(x, x^i)$  is that its magnitude for a stored pattern  $x^i$  be inversely proportional to its distance from the input pattern  $x$  (if the distance is zero the weight is a maximum of unity). The standard distance and weight functions are given by the following two equations, respectively [10]:

$$D(x_1, x_2) = \sum_{k=1}^n \left( \frac{x_{1k} - x_{2k}}{\sigma_k} \right)^2 \tag{2}$$

$$W(x, x^i) = e^{-D(x, x^i)} \tag{3}$$

In (2), each input variable has its own sigma value ( $\sigma_k$ ). This formulation is different from Specht's [2] original work where he used a single sigma value for all input variables. Figure 1 shows a schematic depiction of the four layers GRNN. The first, or input layer, stores an input vector  $x$ . The second is the pattern layer which computes the distances  $D(x, x^i)$  between the incoming pattern  $x$  and stored patterns  $x^i$ . The pattern nodes output the quantities  $W(x, x^i)$ . The third is the summation layer. This layer computes  $N_j$ , the sums of the products of  $W(x, x^i)$  and the associated known output component  $y_i$ . The summation layer also has a node to compute  $D$ , the sum of all  $W(x, x^i)$ . Finally, the fourth layer divides  $N_j$  by  $D$  to produce the estimated output component  $y'_j$ , that is a localized average of the stored output patterns [10].

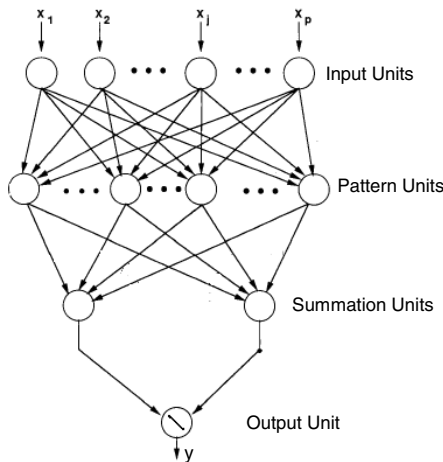


Fig. 1. General Regression Neural Network (GRNN) Architecture

### 3 The Realization of the System for Pattern Recognition

For the object recognition application, each of the 16x16 grayscale input images is converted to one 1x256 input vector. Similarly, for the handwritten digit recognition application, each of the 28x28 grayscale input images is converted to one 1x784 input vector in the designed model and applied to the input of the network. The stages of GRNN layers for the object recognition system are shown in Fig. 2. The stages are the same for the handwritten digit recognition except that the size of the input vector differs.

For recognition process, a code is given for each object as target values of the network. First layer stores reference vectors in GRNN. The distances and weight functions between the unknown vector and each of the stored vectors are computed in the second layer. The distance value computed here shows the rate of the difference between the unknown image and the reference image. As the distance value increases the weight function given in (3) will decrease down to zero. In the third layer, each of  $W$  and the known output components (target values) are multiplied and then added to obtain  $N$ . In the computation of the total value, the smaller  $W$  between unknown input image and reference image means the worse approximation to the target value due to the multiplication of  $W$  and target value of the reference image. Vice versa the greater  $W$  value means better approximation to the target value of the reference image.

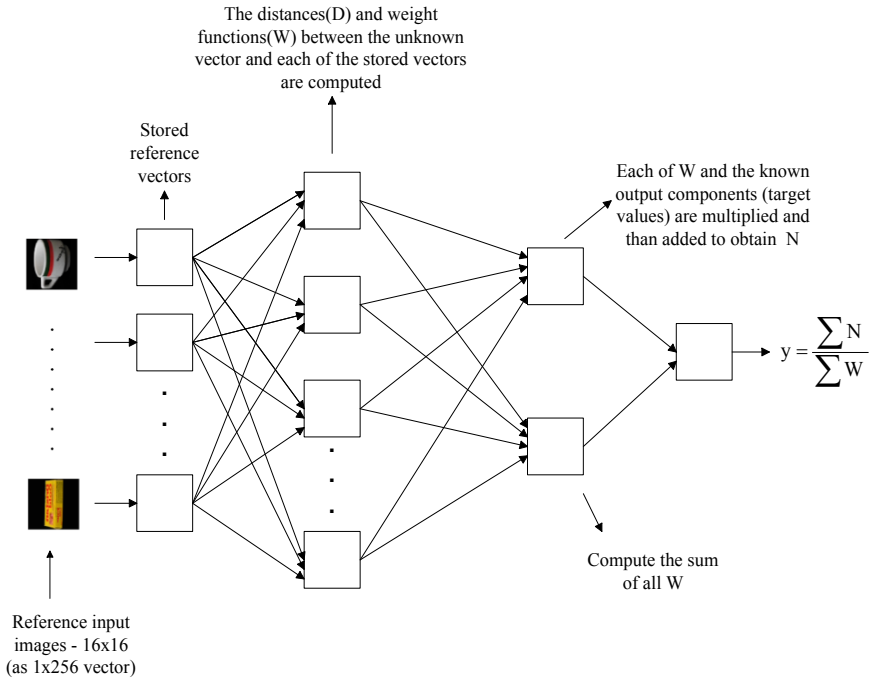


Fig. 2. The stages of GRNN layers for the object recognition system

In the simulation of GRNN for object recognition, different numbers of poses from original data set with different rotation intervals are selected as the reference poses. The remaining images in the data set are used to test the network. For handwritten digit recognition, 50% of the dataset is defined as reference and the system is tested by the remaining 50% of the dataset.

## 4 Simulation and Results

### 4.1 Object Recognition

For the simulations, we are currently using images of the Columbia image database. Columbia Object Image Library (COIL-100) is a database of color images of 100 objects. We selected the 10 objects from the dataset shown in Fig. 3. The objects were placed on a motorized turntable against a black background. The turntable was rotated through 360 degrees to vary object pose with respect to a fixed color camera. Images of the objects were taken at pose intervals of 5 degrees. This corresponds to 72 poses per object. The images were size normalized [11]. Figure 4 shows the frames with rotations  $0^\circ$  to  $25^\circ$  in the object-6 dataset from COIL100.



**Fig. 3.** 10 objects used the simulate recognition system



**Fig. 4.** The image sequence of object-6 in database with rotations  $0^\circ$  to  $25^\circ$

6 and 12 poses from the original dataset with  $60^\circ$  and  $30^\circ$  rotation intervals, respectively, are selected as the reference poses as seen in Table 1. The remaining images in the dataset are used to test the network.

**Table 1.** Number of reference images and corresponding recognition rates

Number of reference images	Recognition rate of the test set
6 poses from the original data set with 60° rotation interval are selected as the reference poses. The remaining 66 images in the data set are used to test the network.	85,15 %
12 poses from the original data set with 30° rotation interval are selected as the reference poses. The remaining 60 images in the data set are used to test the network.	95,83 %

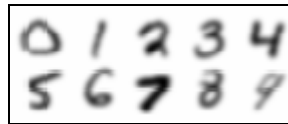
In Table 2, the accuracy of the test results obtained by using 12 poses taken with 30° rotation intervals for the ten objects considered are given. As can be seen from Table 2, recognition rate is quite high for each object.

**Table 2.** Recognition rates of objects

	Obj1	Obj2	Obj3	Obj4	Obj5	Avarage
<b>Test results</b>	86,6%	100%	93,3%	100%	100%	95,83 %
	<b>Obj6</b>	<b>Obj7</b>	<b>Obj8</b>	<b>Obj9</b>	<b>Obj10</b>	
	90%	100%	88,3%	100%	100%	

## 4.2 Handwritten Digit Recognition

In this work, the total of 200 digits, 20 for each digit, is taken from MNIST handwritten digit set (It is a subset of a larger set available from NIST) [12]. The digits have been size-normalized and centered in a fixed-size image. The recognition is done without any feature extraction and preprocessing stages for the digits in the data set. Ten samples for each of the digits between 0 – 9, hence 100 samples, is used for training, a different set of ten samples for each of the digits is used for test. Gray level images of some of the digits between 0 – 9 in the data set is shown in Fig. 5.

**Fig. 5.** Digits from the MNIST-Database

Each of the 28x28 input images is converted to one 1x784 input vector in the designed model and applied at the input of the network. The vectors selected as reference from the data set are applied at the input of GRNN network. A code is given for

each digit as target values of the network and the network is simulated. The network is tested with the rest of the dataset. The test results for 10 references and 10 test data for each digit is presented in Table 3.

**Table 3.** Recognition rates of handwritten digits

Digits	0	1	2	3	4	5	6	7	8	9	Ava.
Test results	100%	70%	70%	80%	20%	60%	50%	60%	60%	80%	65 %

The training rate of the network is 100%. The network response for test data is low for only 4 digits. The network has high test rate for digits 0, 3 and 9. An average test rate is obtained for the other input digits. These test rates can be increased using more samples in the reference set.

The designed model which uses GRNN network has 100% training and 95.8% test performance for object recognition. The training time of the network in a P4 3GHZ, 256 MB RAM PC is 0.07 seconds.

The same application is done using several different neural networks. The Radial Basis Function (RBF) neural network has high training performance whereas test performance is around 60%. The MLP network is also simulated using different training algorithms. There is memory shortage problem for 28x28 size input image for handwritten character recognition application. The input image is resized as 16x16 for the object recognition application. The MLP used in this application has one hidden layer having ten neurons. It takes 8 minutes to train this network to achieve minimum error rate. The resulting network has high training performance but the test performance is around 75%. These simulations indicate that proposed GRNN network has both the speed and accuracy advantage for test data over those of MLP and RBF.

## 5 Conclusion

In this work, a new system is designed for pattern recognition using GRNN. One of the important properties of the proposed system is that there is no need for the pre-processing and feature extraction stages other than image resizing. System was tested for 3D object recognition using 2D poses and handwritten digits recognition. For object recognition, the application is carried out for 10 objects and high recognition rate is obtained. The ability of recognizing unknown objects after training with low number of samples is another important property of this system. For handwritten digits recognition, the application is carried out for 10 digits and considerable recognition rate is obtained. However, the number of the data chosen as reference can be increased in this application.

The availability of different input patterns shows that the system is able to be used in pattern recognition applications such as handwritten character recognition and object recognition. This topology can be applied for pattern recognition while



avoiding sophisticated preprocessing and feature extraction. Furthermore, the dynamic structure of the GRNN network results in faster training. Simulation results show that pattern recognition by GRNN improves the recognition rate considerably in comparison to other neural network structures and has shown better recognition rates and much faster training times than that of Radial Basis Function and Multilayer Perceptron neural networks.

## References

1. Bao-Qing Li; Baoxin Li; Building pattern classifiers using convolutional neural networks. International Joint Conference on Neural Networks, IJCNN'99. Vol. 5, 10-16 July 1999 Page(s):3081 – 3085.
2. D. F. Specht : A general regression neural network. IEEE Trans. Neural Networks, vol. 2, pp. 568–576, (Nov. 1991).
3. D. F. Specht,: Enhancements to probabilistic neural network. in Proc.Int. Joint Conf. Neural Network, vol. 1, pp. 761–768, (1991).
4. Lian-Wei Zhao; Si-Wei Luo; Ling-Zhi Liao; 3D object recognition and pose estimation using kernel PCA. Proceedings of 2004 International Conference on Machine Learning and Cybernetics, Vol. 5, 26–29 Aug. 2004 Page(s):3258 – 3262.
5. Okamoto, J., Jr.; Milanova, M.; Bueker, U.; Active perception system for recognition of 3D objects in image sequences. International Workshop on Advanced Motion Control - AMC'98. Coimbra., 29 June-1 July 1998, Page(s):700 – 705.
6. Polat O.,Tavsanoğlu V.; 3-D Object recognition using 2-D Poses Processed by CNNs and a GRNN. Lecture Notes in Artificial Intelligence, vol.3949, pp.219-226, Springer-Verlag, July, 2006.
7. Nebauer, C.; Evaluation of convolutional neural networks for visual recognition. IEEE Transactions on Neural Networks, Vol. 9, Issue 4, July 1998 Page(s):685 – 696.
8. Fasel, B.; Head-pose invariant facial expression recognition using convolutional neural networks. Fourth IEEE International Conference on Multimodal Interfaces, 14-16 Oct. 2002 Page(s):529 – 534.
9. Simard, P.Y.; Steinkraus, D.; Platt, J.C.; Best practices for convolutional neural networks applied to visual document analysis. Seventh International Conference on Document Analysis and Recognition. 3-6 Aug. 2003, Page(s):958 – 963.
10. Heimes, F.; van Heuveln, B.; The normalized radial basis function neural network. 1998 IEEE International Conference on Systems, Man, and Cybernetics. Vol.2, 11-14 Oct. 1998 Page(s):1609 – 1614.
11. Sameer A. Nene , Shree K. Nayar , Hiroshi Murase: Columbia Object Image Library (COIL-100). Technical Report No. CUCS-006-96, Department of Computer Science Columbia University New York, N.Y. 10027.
12. Available in <http://www.yapay-zeka.org/modules/news/article.php?storyid=3>

# Real-Time String Filtering of Large Databases Implemented Via a Combination of Artificial Neural Networks

Tatiana Tambouratzis

Department of Industrial Management and Technology, University of Piraeus,  
107 Deligiorgi St., Piraeus 185 34, Athens, Greece  
tatianatambouratzis@gmail.com  
<http://www.tex.unipi.gr/dep/tambouratzis/main.htm>

**Abstract.** A novel approach to real-time string filtering of large databases is presented. The proposed approach is based on a combination of artificial neural networks and operates in two stages. The first stage employs a self-organizing map for performing approximate string matching and retrieving those strings of the database which are similar to (i.e. assigned to the same SOM node as) the query string. The second stage employs a harmony theory network for comparing the previously retrieved strings in parallel with the query string and determining whether an exact match exists. The experimental results demonstrate accurate, fast and database-size independent string filtering which is robust to database modifications. The proposed approach is put forward for general-purpose (directory, catalogue and glossary search) and Internet (e-mail blocking, intrusion detection systems, URL and username classification) applications.

## 1 Introduction

The aim of string comparison [1-3] is to retrieve appropriately selected strings from a given database. String comparison is widely used not only in text processing but also in molecular biology [4], computer science, coding and error control, pattern matching and handwritten text/character recognition [5-7], natural language processing, music comparison and retrieval [3,8-9] etc. Assuming a database of strings and a query string, string comparison falls into two main categories:

- Perfectly matching (entire or parts of) strings with the query string [2,10-13]. The aim is to locate the occurrences of the query string in the database, i.e. to extract all the database strings that are identical, or contain substrings that are identical, to the query string. Two interesting aspects of perfect string matching are worth mentioning. The first is relevant to molecular biology and requires that the query string appears as a substring at similar locations of the database strings [4]; such strings share structural similarities and, thus, belong to the same family. The second, string filtering, is applicable to databases of discrete strings (e.g. URLs, usernames, telephone number catalogues, glossaries and other word lists) and

determines whether the query string exists in (i.e. is identical to an entire string of) the database [14].

- Approximately matching (entire or parts of) strings of the database with the query string [1,3,5-9,15-25]. The task is to retrieve strings which are similar, or contain substrings that are similar, to the query string; string similarity is expressed via special operators (e.g. Levenshtein distance [26], string block edit distance, longest common subsequence) that detect not only substring permutations but also character<sup>1</sup> insertion, deletion and substitution errors. Branch-and-bound-search [8], automata theory [21-22], genetic algorithms [7] etc. have been introduced for efficiently solving approximate string searching and matching.

String filtering has been established as the means of identifying/verifying/authenticating query strings in telephone-number and/or name catalogues, glossaries, word lists etc. Furthermore, with the steady development of the Internet and the growing usage of its applications, string filtering and blocking are becoming increasingly important for classifying and accepting/blocking URLs, e-mail addresses/messages etc., where the query strings are suitably selected keywords combined according to certain rules. A string filtering example, which is crucial for preventing attacks over the Internet, is intrusion detection systems (IDS) and especially network IDS (NIDS); owing to the fact that the databases of both keywords and URLs, e-mail addresses, e-mail messages etc. are growing rapidly, string filtering must not only be foolproof, but it must also remain practically database-size independent as well as robust to frequent database modifications (additions/deletions of key strings and/or combination rules).

## 2 Traditional String Filtering

Traditionally executed string filtering requires that the database, which is assumed to comprise  $N$  strings of average length  $M$ , has been sorted in alphabetical order<sup>2</sup>. Given the query string, string filtering is initialised by comparing the query string with the central (e.g. the  $\text{CEIL}(N/2)+1$ th, where  $\text{CEIL}$  is the ceiling operator) string of the sorted database. If successful, search terminates with a "match" notification. Otherwise, search is directed towards the first/second portion of the sorted database, depending on whether the query string alphabetically precedes/follows the unsuccessfully matched string of the database. Database division and central string filtering are repeated until either a perfect match is found (whereby a "match" notification is returned) or the portion of the database towards which matching is directed becomes empty (whereby a "no match" notification is returned). Although 100% accurate, this kind of filtering is clearly database size-dependent, with its computational complexity equaling  $O(\log_2(N))$  plus the  $M$ -dependent overload for string/query string matching. Furthermore, any modification of the database requires database reordering. Despite the use of indexing [27-29] for speeding up filtering, a

<sup>1</sup> The elementary component of a string of any kind.

<sup>2</sup> Sorting can also be performed according to some other criterion of string ordering, e.g. ASCII-based ordering.

speed-memory overload trade-off is created; hybrid sorting [30] and coding techniques [31] have also been put forward for alleviating these problems to some extent.

### 3 Proposed String Filtering Approach

The proposed string filtering approach is based on a combination of artificial neural networks (ANNs) and operates in two stages:

- The first (approximate matching) stage employs a self-organizing map (SOM) [32] for grouping together similar strings of the database. Clustering is accomplished by collecting all the database strings assigned to the same SOM node, i.e. sharing the same best-matching unit (BMU). String lists are collected over all SOM nodes, where the lists constitute a partition of the database. Given a query string, its BMU is determined and the list of database strings assigned to the same BMU is retrieved.
- The second (perfect matching) stage employs a harmony theory network (HTN) [33] for determining whether a perfect match exists between the query string and the previously retrieved (list of) strings. Matching is performed character-by-character and in parallel over the retrieved strings of the database. Once a mismatch occurs between a retrieved string and the query string, this string of the database is no longer considered for matching. Matching is successful only if exactly one retrieved database string is found to match the query string up to the last character.

A maximum allowable length  $L$  for the strings in the database is assumed, whereby strings with fewer characters are padded with the appropriate number of blank characters. The padded strings are converted into sequences of numbers, with each character of the string represented by its corresponding number; either ASCII or any other appropriately selected set of numbers can be used. String similarity is measured in terms of the overall (collected over the  $L$  characters) number-proximity of converted characters at corresponding locations of the strings; in other words, the more corresponding characters with highly similar numbers, the stronger the similarity between strings. The free choice of character-number conversion offers extended flexibility in expressing string similarity. For instance, using

- similar numbers for characters located nearby in the keyboard accommodates for typing errors,
- distinct numbers for desired (e.g. special) characters sets apart words with and without these characters,
- the same numbers to lower- and upper-case letters provides case-insensitivity etc.

This is especially useful in case of unsuccessful string filtering, should it be of interest to also determine the database strings that best (even if only approximately) match the query string.

The ANNs involved in the two stages are detailed next.

### 3.1 Approximate Matching

During the first stage, all the strings of the database are converted, normalized and, subsequently, employed for training a 2-D SOM: the mapping from the  $L$ -dimensional input data onto a two-dimensional array of nodes preserves the most important topological and metric relationships of the data so that similar strings are grouped together. The SOM dimensions are determined according to the minimization of topological and quantization errors<sup>3</sup> during training.

Directly after SOM training, the strings of the database are input into the SOM one-by-one and their BMUs are determined; this results into each SOM node being assigned a list of "similar" strings.

It is these lists that are employed for performing approximate string matching with a query string. The query string is input into the SOM and its BMU is determined. The strings of the database assigned to this node constitute the strings which are retrieved as being similar to the query string, i.e. the database strings that are candidate for exact matching.

It should be mentioned that, although relatively large and sparse SOMs are created by the aforementioned training procedure (requiring longer training and extra computational effort during testing), a small number of strings is assigned to each SOM node (i.e. list of similar strings).

### 3.2 Exact Matching

The second stage constitutes a variant of the simplified HTN employed for solving a variety of on- and off-line string matching as well as sorting problems [34-35]. The query string is matched character-by-character and in parallel over all the strings previously retrieved by the SOM.

An ensemble of HTN structures is utilized, where each HTN structure:

- is allocated to a string of the database retrieved during the previous stage, and
- comprises a single node in the lower layer, a single node in the upper layer and a fixed weight of +1 between them.

Exact matching proceeds in - at most -  $L$  time steps, where for the first time step,

- the nodes of the lower layer are assigned the numbers resulting from the conversion of the first characters of the allocated strings,
- the number resulting from the conversion of the first character of the query string is input simultaneously into the HTN structures,
- matching and HTN updating are performed according to [34-35] so that the nodes of the upper layer with identical/distinct input and lower layer node numbers are assigned +1/0 values,
- the HTN structures with +1 values of their upper layer node remain active, while the remaining HTN structures become de-activated.

---

<sup>3</sup> The topological error (proportion of strings of the database for which the BMU and the second BMU are not neighbouring nodes) is given priority over the quantization error (the average distance of the strings of the original database from their BMUs) since the former constitutes a measure of map continuity whereas the latter is more sensitive to database modifications.

The next ( $i=2, 3, 4, \dots, L$ ) time steps are performed in the same fashion; the HTN structures which proceed to the  $i$ th time step are those whose allocated strings share the first  $i-1$  characters with those of the query string. HTN operation resembles sliding the input characters of the query string concurrently with the corresponding characters of the previously retrieved - and so-far matching - strings over the HTN structures, matching in parallel and calculating the activity/de-activation status of the HTN structures. Matching ceases either as soon as no active HTN structures remain (whereby a "no match" notification is returned) or when sliding the query string has been completed and a single HTN structure remains active (whereby a "match" notification is returned). Since fewer HTN structures remain active at each subsequent time step, the computational complexity of exact matching is reduced at each character of the query string<sup>4</sup>.

### 3.3 Database Updating

When a string is removed from the database, it is also removed from the list of its BMU. Conversely, when a new string is to be added to the database, it is input into the SOM and its BMU is determined; the new string is appended to the list of strings assigned to this node as well as to the database only if exact matching of the new string with the strings already assigned to the same BMU fails, i.e. the new string is distinct from the strings of the database.

This updating procedure works under conditions of extensive database modifications: since the SOM characteristics remain invariable, the BMUs of the strings - either original or added later on in the updating process - do not change. In order, however, to maintain efficient operation, SOM training is repeated anew (concerning both SOM topology and node weights) either (i) once 20-25% of the strings of the original database have changed<sup>5</sup> or (ii) when the longest list of strings assigned to SOM nodes (of size  $K$ ) has grown by 50% (i.e.  $K=K^*=\text{CEIL}(1.5K)$ )<sup>6</sup>.

The replication of set-up secures effective operation: the SOM dimensions are appropriately adjusted so that the sparseness and continuity of mapping (between database strings and SOM nodes) are preserved. The added time and space complexity of SOM training (especially after database expansion, whereby larger SOMs are constructed) is counterbalanced by the preservation of limited cluster sizes and, thus, cost-effective exact matching.

## 4 Demonstration of String Filtering

An example of string filtering is presented for the database containing the usernames (of length between 2 and  $L=11$  characters prior to padding) of the faculty members of

<sup>4</sup> Swift computational complexity reduction at each time step is also promoted by the criterion of overall string similarity: although similar strings are clustered together, the characters of strings at corresponding locations are not necessarily the same.

<sup>5</sup> This constitutes the most commonly applied criterion for replicating set-up.

<sup>6</sup> It has been observed that longer lists of strings tend to be enriched with new strings at a faster rate than shorter lists. In fact,  $K^*$  constitutes that number of reserved HTNs employed by the proposed approach for exact string matching; hence, at the point where the longest list of strings contains  $K^*$  strings, it becomes necessary to reconstruct the size of the HTN ensemble.

the University of Piraeus. The character-number correspondence selected for this demonstration is shown in Table 1.

The database is initialized with the faculty member usernames ( $N_{initial}=197$ ) and is progressively enlarged to include the usernames of PhD students and other supporting/affiliated personnel ( $N_{final}=1543$ ). During database updating, usernames are randomly added and removed from the database at a rate of one removal for every ten additions. After each database modification (a total of 1481 modifications, namely 1346 string additions and 135 string removals), the proposed approach is tested with the entire database ( $N_{final}=1453$  strings); while many mismatches and few matches are observed at first, the situation progressively reverses. Database updating (from the initial to the final database) has been repeated 25 times; the averaged results over the 25 tests are described next. The SOMTOOLBOX for the MATLAB environment has been used [36] for approximate matching (SOM construction, approximate string filtering and set-up replication).

**Table 1.** Character-number conversion

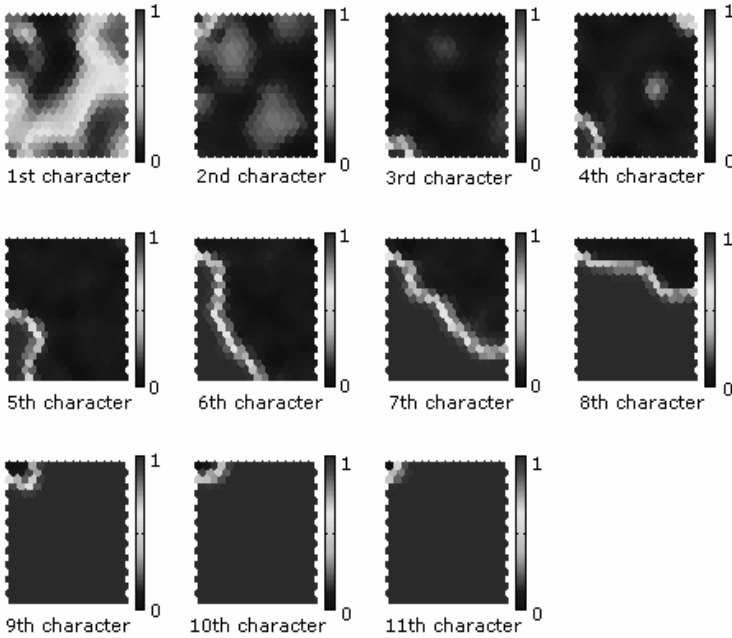
Character	Number
a-z	1-26
A-Z	31-56
1-0	71-80
Special (__, -, /, ., etc.)	85-96
blank	200

For the initial set-up, a 20x15 hexagonal-lattice SOM has been found to simultaneously minimize the quantization and topological errors (average of 0.6673 and 0, respectively); linear weight initialization, independent normalization per dimension (character location), a Gaussian neighborhood (with a gradual shrinkage from the original radius 3 to a final radius of 1) and batch training with an initial rough phase of 64 epochs and a fine-tuning phase of 244 epochs have been employed. Fig. 1 illustrates the weights of the 300 nodes per dimension after the initial SOM set-up, emphasizing the priority given to clustering by string length<sup>7</sup>. The sparseness of the SOM is exemplified in Table 2: most nodes have void clusters, while few strings are assigned to each non-empty list of strings. Subsequently, exact string matching is performed. If the BMU of the query string has an empty list of strings, a "no match" notification is directly reported<sup>8</sup>; otherwise, an ensemble of  $K^*=11$  HTN structures is implemented for exact string matching. Of them, only as many HTNs are initialized to active as there are database strings in the list. This can be seen in Fig. 2 for the second step of exact string matching, where the 8-11th HTNs have been de-activated from the beginning (assuming that the list of retrieved strings comprises seven strings,

<sup>7</sup> This is accomplished by having assigned a highly distinct number to the blank character (Table 1).

<sup>8</sup> Assuming that the query string can be assigned to any SOM node with equal probability, the probability of a direct "no match" response equals 0.6033.

which constitutes the worst possible case as shown Table 2) and the 2nd HTN was de-activated during the first time step due to a mismatch; the de-activated HTNs have been marked in Fig. 2 as black nodes of the upper layer.



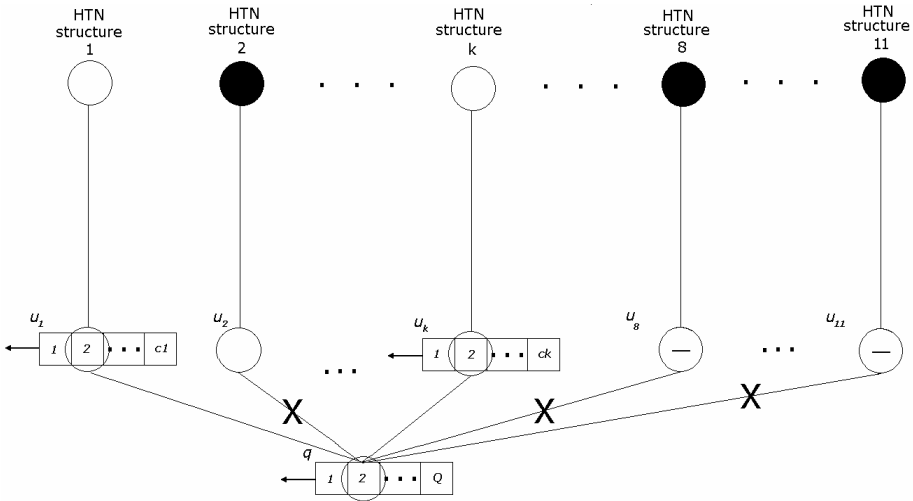
**Fig. 1.** SOM weights per dimension (character): demonstration of section 4

In terms of computational complexity,

- approximate string matching requires 11 number comparisons (similarity per dimension) and 10 additions (overall string-node similarity) for determining string similarity with each SOM node, followed by BMU selection among the 300 nodes;
- for exact matching, a "no match" notification is directly reported (i.e. the list of strings corresponding to the BMU is empty) in 82.52% of the cases. In the remaining cases (non-empty lists of strings), an average of 1.26 HTNs are initialized to active. Inspection of the strings in the non-empty lists of strings reveals:
  - One pair of strings in the same list matching up to the sixth character;
  - One pair of strings in the same list matching up to the fourth character;
  - Five pairs and one triplet of strings in the same list matching up to the second character;
  - 14 pairs and one triplet of strings in the same list matching up to the first character,

whereby swift HTN de-activation is demonstrated.





**Fig. 2.** HTN structures for exact matching: demonstration of section 4, second step

In all 25 tests, it has been found necessary to repeat set-up ten times during the progression of database updating (from  $N_{initial}$  to  $N_{final}$ ); both the SOM dimensions and the size of the HTN ensemble gradually grow. A 40x28 SOM has always been constructed for the final set-up, demonstrating comparable training errors, SOM characteristics and cluster sizes (ranging in size between 0 and 12 and similarly distributed as those in Table 2) to those of the initial set-up; 18 HTNs constitute the final ensemble for exact string matching. During the entire operation and set-up replication, the proposed approach has been found foolproof, fast and practically database-independent.

**Table 2.** BMU list sizes

# Nodes with	# Assigned strings
181	0
71	1
31	2
10	3
3	4
3	5
-	6
1	7

## 5 Conclusions

A novel approach to real-time string filtering of large databases has been presented. This is based on a combination of artificial neural networks and operates in two

stages. The first (approximate string matching) stage employs a self-organizing map for clustering similar strings of the database; given a query string, only those strings of the database whose best-matching unit coincides with that of the query string are retrieved. The second (exact string filtering) stage employs an ensemble of harmony theory networks for determining whether a perfect match exists between the previously retrieved strings and the query string. Matching is performed character-by-character, in parallel over the retrieved strings of the database. Once a mismatch occurs with a retrieved string, this is no longer considered for matching; as a result, the computational complexity of matching is reduced at each subsequent character of the query string. The experimental results demonstrate accurate, fast and database-size independent string filtering, which is robust to database modifications.

## References

1. Boyer, R., Moore, S.: A Fast String Matching Algorithm. *Comm. ACM* 20 (1977) 762-772
2. Knuth, D.E., Morris, J., Pratt, V.: Fast Pattern Matching Strings. *SIAM J. Comp.* 6 (1977) 323-350
3. Makinen, V., Navarro, G., Ukkonen, E.: Transposition Invariant String Matching. *J. Algor.* 56 (2005) 124-153
4. Elloumi, M.: Comparison of Strings Belonging to the Same Family. *Inform. Sci.* 111 (1998) 49-63
5. Pao, D.C.W., Sun, M.C., Lam, C.H.: An Approximate String Matching Algorithm for on-Line Chinese Character Recognition. *Im. Vis. Comp.* 15 (1997) 695-703
6. Lopresti, D., Tomkins, A.: Block Edit Models for Approximate String Matching. *Theoret. Comp. Sci.* 181 (1997) 159-179
7. Parizeau, M., Ghazzali, N., Hebert, J.F.: Optimizing the Cost Matrix for Approximate String Matching Using Genetic Algorithms. *Patt. Recogn.* 32 (1998) 431-440
8. Lemstrom, K., Navarro, G., Pinzon, Y.: Practical Algorithms for Transposition-Invariant String-Matching. *J. Discr. Alg.* 3 (2005) 267-292
9. Deodorowicz, S.: Speeding up Transposition-Invariant String Matching. *Inform. Proc. Lett.* 100 (2006) 14-20
10. Crochemore, M., Gasieniec, L., Rytter, W.: Constant-Space String-Matching in Sublinear Average Time. *Theor. Comp. Sci.* 218 (1999) 197-203
11. Misra, J.: Derivation of a Parallel String Matching Algorithm. *Inform. Proc. Lett.* 85 (2005) 255-260
12. Allauzen, C., Raffinot, M.: Simple Optimal String Matching Algorithm. *J. Alg.* 36 (2000) 102-116
13. He, L., Fang, B., Sui, J.: The Wide Window String Matching Algorithm. *Theor. Comp. Sci.* 332 (2005) 301-404
14. Ramesh, H., Vinay, V.: String Matching on  $\tilde{O}(\sqrt{n} + \sqrt{m})$  quantum time. *J. Discr. Alg.* 1 (2003) 103-110
15. Horspool, R.: Practical Fast Searching in Strings. *Oft. Pract. & Exper.* 10 (1980) 501-506
16. Sunday, D.M.: A very Fast Substring Search Algorithm. *Comm. ACM* 33 (1990) 132-142
17. Galil, Z., Park, K.: An Improved Algorithm for Approximate String Matching. *SIAM J. Comp.* 19 (1990) 989-999
18. Baeza-Yates, R.A., Perleberg, C.H.: Fast and Practical Approximate String Matching. *Inf. Proc. Lett.* 59 (1996) 21-27

19. Landau G., Vishkin U.: Fast String Matching with k Differences, *J. Comp. Sys. Sci.* 37 (1988) 63-78
20. Navarro, G., Baeza-Yates, R.: Very Fast and Simple Approximate String Matching. *Inf. Proc. Lett.* 72 (1999) 65-70
21. Holub, J., Melichar, B.: Approximate String Matching Using Factor Automata. *Theor. Comp. Sci.* 249 (2000) 305-311
22. Choffrut, Ch., Haddad, Y.: String-Matching with OBDDs. *Theor. Comp. Sci.* 320 (2004) 187-198
23. Hyyro, H.: Bit-Parallel Approximate String Matching Algorithms with Transposition. *J. Discr. Alg.* 3 (2005) 215-229
24. Navarro, G., Chavez, E.: A Metric Index for Approximate String Matching. *Theor. Comp. Sci.* 352 (2006) 266-279
25. Nebel, M.E.: Fast String Matching by Using Probabilities: an Optimal Mismatch Variant of Horspool's Algorithm. *Theor. Comp.* 359 (1006) 329-343
26. Levenshtein, A.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Sov. Phy. Dokl.* 10 (1966), 707-710
27. Manber, U., Myers, E.W.: Suffix Arrays: a New Method for On-Line String Searches. *SIAM J. on Comp.* 22 (1993) 935-948
28. Moffat, A., Zobel, J.: Self-Indexing Inverted Files for Fast Text Retrieval. *ACM Trans. Inf. Sys.* 14 (1996) 349-379
29. Ferragina, P., Grossi, R.: The String B-Tree: a New Structure for String Search in External Memory and Application. *J. of ACM* 46 (1999) 236-280
30. Bentley, J., Sedgewick, R.: Fast Algorithms for Sorting and Searching Strings. *Proc. Of the ACM-SIAM Symposium on Discrete Algorithms* (1997) 360-369
31. Grossi, R., Vitter, J.S.: Compressed Suffix Arrays and Suffix Trees with Applications to Text Indexing and String Matching. *Proc. Of the 3<sup>rd</sup> Annual ACM Symposium on Theory of Computation* (2000) 397-406 (also in *SIAM J. on Comp.* 35 (2005))
32. Kohonen, T.: *Self-Organizing Maps* (3<sup>rd</sup> edition). Springer-Verlag, Berlin Heidelberg, Germany (2001).
33. Smolensky, P.: Information Processing in Dynamical Systems: Foundations of Harmony Theory. In: Rumelhart, D.E., McClelland, J.L. (eds.): *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge MA (1986) 194-281
34. Tambouratzis, T.: String Matching Artificial Neural Networks. *Int. J. Neur. Syst.* 11 (2001) 445-453
35. Tambouratzis, T.: A Novel Artificial Neural Network for Sorting. *IEEE Trans. Syst., Man & Cybern.* 29 (1999) 271-275
36. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: *SOM Toolbox for Matlab 5*. Report A57 (2000). SOM Toolbox Team, Helsinki University of Technology, Finland (available at <http://www.cis.hut.fi/projects/somtoolbox>)

# Parallel Realizations of the SAMANN Algorithm

Sergejus Ivanikovas, Viktor Medvedev, and Gintautas Dzemyda

Institute of Mathematics and Informatics, Akademijos 4, LT-08663, Vilnius, Lithuania  
Ivanikovas@gmail.com, Viktor.m@ktl.mii.lt, Dzemyda@ktl.mii.lt

**Abstract.** Sammon's mapping is a well-known procedure for mapping data from a higher-dimensional space onto a lower-dimensional one. But the original algorithm has a disadvantage. It lacks generalization, which means that new points cannot be added to the obtained map without recalculating it. The SAMANN neural network, that realizes Sammon's algorithm, provides a generalization capability of projecting new data. A drawback of using SAMANN is that the training process is extremely slow. One of the ways of speeding up the neural network training process is to use parallel computing. In this paper, we proposed some parallel realizations of the SAMANN.

## 1 Introduction

Searches for suitable and better data projection methods have always been an integral objective of pattern recognition and data analysis. Feature extraction is the process of mapping the original features into a smaller amount of features, which preserve the main information of the data structure. Such visualizations are useful especially in exploratory analyses: they provide overviews of the similarity relationships in high-dimensional datasets that would be hardly obtainable without the visualization.

The problem of data projection is defined as follows: given a set of high-dimensional data points (vectors), project them to a low-dimensional space so that the resulting configuration would perform better than the original data in further processing, such as clustering, classification, indexing and searching [7, 9]. In general, this projection problem can be formulated as mapping a set of  $n$  vectors from the  $d$ -dimensional space onto the  $m$ -dimensional space, with  $m < d$ .

The classical method for projecting data is Multi-Dimensional Scaling (MDS) [2]. This method works with inter-point distances and gives a low-dimensional configuration that represents the given distances best. One of the popular MDS-type projection algorithms is Sammon's method [14].

Mao and Jain [8, 10] have suggested the implementation of Sammon's mapping by a neural network. A specific backpropagation-like learning rule has been developed to allow a normal feedforward artificial neural network to learn Sammon's mapping in an unsupervised way. The neural network training rule of this type was called SAMANN. In Mao and Jain's implementation, the network is able to project new multidimensional points (vectors) after training – the feature missing in Sammon's mapping. Projection of new point sets is investigated in [11].

A drawback of using SAMANN is that it is rather difficult to train and the training process is extremely slow. One of the ways of speeding up the neural network training process is to use parallel computing. In this paper, we proposed some parallel realizations of the SAMANN to reduce the computation time of the algorithm.

Parallel programming is widely used while working with super computers and clusters. Parallel computing allows us to process one task in different places at the same time and increase the computation speed. Nowadays clusters, SMP (Symmetric Multi-Processor) systems, and computers with HT (Hyper-Threading) technology have become popular and available for most users [3], [15]. The parallel realization of the SAMANN can solve the problem of slow neural network learning.

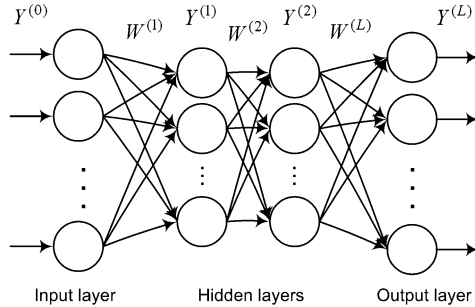
## 2 A Neural Network for Sammon's Projection

Sammon's nonlinear mapping is an iterative procedure to project high-dimensional data  $X = (x_1, x_2 \dots, x_d)$  into low-dimensional configurations  $Y = (y_1, y_2 \dots, y_m)$ , where  $m < d$ . It tries to keep the same interpattern distances between points in the low-dimensional space. Suppose that we have  $n$  data points,  $X_i = (x_{i1}, x_{i2} \dots, x_{id})$ ,  $i = 1, \dots, n$ , in a  $d$ -dimensional space and, respectively, we define  $n$  points,  $Y_i = (y_{i1}, y_{i2} \dots, y_{im})$ ,  $i = 1, \dots, n$ , in a  $m$ -dimensional space ( $m < d$ ). Without loss of generality, only projections onto a two-dimensional space are studied ( $m = 2$ ). The pending problem is to visualize these  $d$ -dimensional vectors  $X_i$  onto the plane  $R^2$ . Let  $d_{ij}^*$  denote the Euclidean distance between  $X_i$  and  $X_j$  in the input space and  $d_{ij}$  denote the Euclidean distance between the corresponding points  $Y_i$  and  $Y_j$  in the projected space. The projection error measure  $E$  (Sammon's error) is as follows:

$$E = \frac{1}{\sum_{\substack{i,j=1 \\ i < j}}^n d_{ij}^*} \sum_{\substack{i,j=1 \\ i < j}}^n \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*} .$$

Since Sammon's algorithm was primarily designed for data analysis and visualization, one of its major drawbacks is that it does not yield a map or algorithm that might allow one to generalize the transformation to unseen points. The SAMANN for Sammon's nonlinear projection was a neural network training paradigm proposed in [10]. This algorithm allows us to avoid problem, mentioned above. The SAMANN network for  $m$ -dimensional projection is given in Fig. 1. It is a feedforward neural network where the number of input units is set to be the feature space dimension  $d$ , and the number of output units is specified as the extracted feature space dimension  $m$ .  $Y^{(k)} = \{y_i^{(k)}\}$  is a vector of outputs of the  $k$ -th layer of neural network,  $Y^{(0)} = X = (x_1, x_2, \dots, x_d)$ .  $W^{(k)} = \{w_{ij}^{(k)}\}$  is a matrix of weights of the  $k$ -th layer of neural network. The weight of connection between unit  $i$  in layer  $k-1$  and unit  $j$  in layer  $k$  is represented by  $w_{ij}^{(k)}$ . Mao and Jain [10] have derived a weight updating rule for the multilayer perceptron that

minimizes Sammon’s error, based on the gradient descent method. The sigmoid activation function with the range (0.0, 1.0) is used for each unit. However, in the neural network implementation of Sammon’s mapping the errors in the output layer are functions of the interpattern distances.



**Fig. 1.** SAMANN network for two-dimensional projection

The network takes a pair of input vectors each time in the training. The outputs of each neuron are stored for both points. The distance between the neural network output vectors can be calculated and an error measure can be defined in terms of this distance and the distance between the points in the input space. From this error measure, a weight update rule has been derived in [10].

The SAMANN Unsupervised Backpropagation Algorithm is as follows:

1. Initialize the weights in the SAMANN network randomly.
2. Select a pair of vectors randomly, present them to the network one at a time, and evaluate the network in a feedforward fashion.
3. Update the weights in the backpropagation fashion starting from the output layer.
4. Repeat steps 2–3 a number of times.
5. Present all the vectors and evaluate the outputs of the network; compute Sammon’s error; if the value of Sammon’s error is below a prespecified threshold or the number of iterations (from steps 2–5) exceeds the prespecified maximum number, then stop; otherwise, go to step 2.

### 3 Strategies of the SAMANN Algorithm Parallelization

Working with SAMANN, we need many computation resources to train the neural network. The training process takes a considerable amount of time [13]. One of the ways of solving this problem and decreasing the computation time is to use parallel computing and adapt the methods of making the consecutive algorithms parallel. This paper shows two strategies of modifying the consecutive algorithm, which allow us to use several processors for the net training at the same time. In this case, the parallel algorithm makes it possible to solve

the given task faster than it could be done using only one processor. In the first strategy, the training data management was parallelized. In the second strategy, the training process was parallelized.

### 3.1 The First Strategy of the Parallel Realization

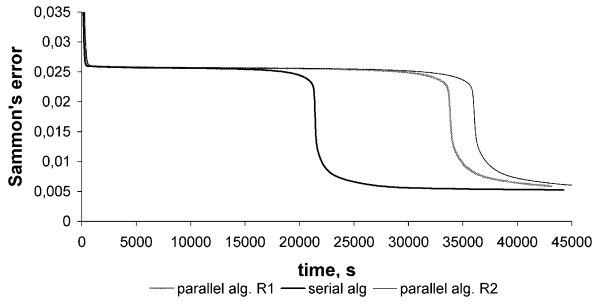
The first strategy operates with three processors, where at least two of them are of identical parameters. However, their number may be enlarged. The neural network is trained by  $d$ -dimensional vector pairs  $(X_i, X_j)$ ,  $i, j = 1, \dots, n, i \neq j$ , using  $M$  iterations. In each training step, two vectors  $\mu$  and  $\nu$  are presented to the neural net (each processor, which implements the calculation of the net weight values, creates two identical neural networks and fulfils all the calculations in its local part). At the beginning of each iteration, the zero processor ( $p_0$ ) randomly mixes the data array  $A = \{X_i = (x_{i1}, x_{i2} \dots, x_{id}), i = 1, \dots, n\}$  and distributes it among the rest two processors  $p_1$  and  $p_2$  (the starting data array elements are divided into equal, or almost equal blocks). Since the starting data array is mixed at the beginning of each iteration, each time the processors get different data blocks of the same size. After the vectors have been delivered to the processors the computation of network outputs starts. These outputs are used for the renewal of the appropriate net weights beginning with the exit layer. All the processors fulfil the calculations in parallel and with local data only. After that stage, each of the two processors has its own local weights  $\overline{w}_{ij}^{(k)}$  and  $\overline{\overline{w}}_{ij}^{(k)}$ , and sends them to the zero processor. The zero processor renews the weights and sends their values back to the other processors. Using the just calculated weights the zero processor calculates all the output vectors of the neural net. After each iteration, the input and output vectors are used for calculation of Sammon's error  $E$ .

For the renewal of net weights, the following rules are used: R1:  $w_{ij}^{(k)} = (\overline{w}_{ij}^{(k)} + \overline{\overline{w}}_{ij}^{(k)})/2$ , so the simple mean is used; R2: from  $\overline{w}_{ij}^{(k)}$  and  $\overline{\overline{w}}_{ij}^{(k)}$  the weight set with a smaller mapping error  $E$  is chosen.

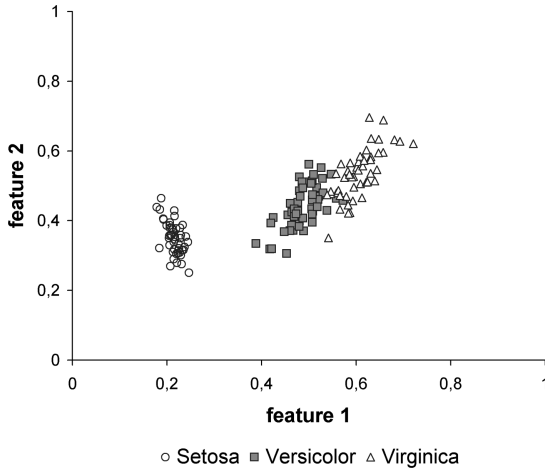
The computer network, controlled by the environment created by MPI (Message Passing Interface) [5], [16] library, was used in the experiment. In the analysis of strategies for the network training, a particular case of the SAMANN network was considered: a feedforward artificial neural network with one hidden layer and two outputs ( $m = 2$ ). In each case, the set of initial weights was fixed in advance. To visualize the initial dataset, the following parameters were employed: the number of neurons of the hidden layer was  $n_2 = 10$  and the learning rate was  $\eta = 0.5$ .

The Iris Dataset was used in the experiments (Fisher's iris dataset) [6]. It is a real dataset with 150 random samples of flowers from iris species: setosa, versicolor, and virginica. The iris flowers were described by 4 attributes ( $d = 4$ ). From each species there were 50 observations of sepal length, sepal width, petal length, and petal width in cm.

The mapping errors were calculated and the calculation time was measured for serial and parallel algorithms. The results of the parallel algorithm are compared with that of the serial one (Fig. 2). Figure 3 illustrates a 2D projection map of the Iris dataset.



**Fig. 2.** Dependence of the projection error on the computation time for the Iris dataset



**Fig. 3.** 2D projection map of the Iris dataset using the SAMANN network

Considering the results received previously, the modified strategy of the parallel realization was suggested. At the beginning of each iteration, the zero processor ( $p_0$ ) randomly mixes the data array  $A = \{X_i = (x_{i1}, x_{i2} \dots, x_{id}), i = 1, \dots, n\}$  and divides it into two equal or almost equal parts  $A_1$  and  $A_2$ , then  $p_0$  distributes it among the rest two processors  $p_1$  and  $p_2$ . Two identical neural networks are created. Each network is trained with appropriate part of data array  $A_1$  or  $A_2$  using  $M_1$  iterations. This process is fulfilled by two processors in parallel, the first processor uses the array  $A_1$  and the second one the array  $A_2$ . Then the calculation is fulfilled by zero processor ( $p_0$ ). After the training ( $M_1$  iterations) where got weight sets  $W_1$  and  $W_2$  of each net, output vectors and appropriate projection errors  $E_1$  and  $E_2$  for whole data array  $A$ . Projection errors  $E_1$  and  $E_2$  are obtained when all the vectors from array  $A$  are used in the nets trained appropriately with data arrays  $A_1$  and  $A_2$ . Of two weight sets remains the one with smaller projection error. Using the obtained weight set the neural network is trained with all data set  $A$  vectors ( $M_2$  iterations).



The above described process can be repeated many times till the Sammon’s error is below a prespecified threshold or the number of iterations exceeds the prespecified maximum number ( $M_3$ ). Each time the arrays  $A_1$  and  $A_2$  must be created over again. This article describes the limited process with out the multiple repeating of the process above.

Two datasets were used in the experiments with the modified strategy:

1. Iris Dataset ( $n = 150, d = 4$ ) [6],
2. Ionosphere Dataset [1]. The Ionosphere dataset contains 351 instances ( $n = 351$ ) and 34 attributes ( $d = 34$ ) representing data gathered from a radar that detects the presence of free electrons in the ionosphere.

To visualize the Iris and the Ionosphere datasets, the following parameters were used: the number of neurons of the hidden layer  $n_2 = 20$  (for the Iris dataset) and  $n_2 = 10$  (for the Ionosphere dataset), the learning rate  $\eta = 5$  and the momentum value of 0.05.

The mapping errors  $E$  were calculated and the calculation time was measured for serial and parallel algorithms. The results of the parallel algorithm are compared with that of the serial one. The result for the Iris dataset and the Ionosphere dataset is presented in Figures 4 and 5.

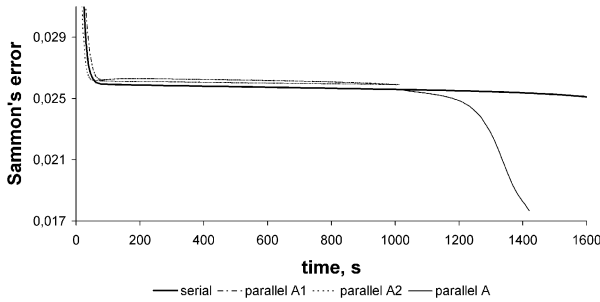


Fig. 4. Dependence of the projection error on the computation time for the Iris dataset

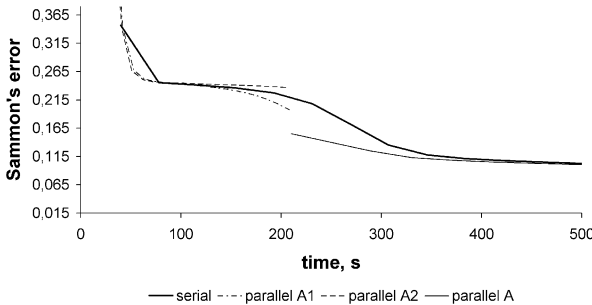


Fig. 5. Dependence of the projection error on the computation time for the Ionosphere dataset

The figures show that the modified parallel algorithm lets to achieve good visualization results during the shorter time. The starting data set  $A$  can be divided into  $k$  parts ( $k \geq 2$ ). Good results were also achieved dividing the data set into 3 parts.

### 3.2 The Second Strategy of the Parallel Realization

When visualizing to the two-dimensional space using SAMANN, it is natural to divide the neural network into two parts (it is also possible to divide the neural network to more than two parts). Each network layer is divided into equal or almost equal parts. Then the learning process of the network can be parallelized. The parallel algorithm that realized this idea is presented below. Such a parallel programme can be used with shared memory computers. The dual processor or dual core SMP systems are popular and available for most users nowadays. The multi-threaded programme can perform faster and more efficiently than the serial one. The usage of this parallel programme with clusters or distributed memory systems is complicated because of data synchronization: the programme will be inefficient because of data transfer between the processors.

The OpenMP standard was used to create a multi-threaded programme. This is one of the most popular ways of programming applications for SMP systems. It allows us to create multi-threaded programmes in a simple and effective way [12], [16].

The proposed parallel algorithm performs network training using two threads. The algorithm divides the SAMANN net training into serial and parallel parts. A part of the computation was performed by one thread (input of the data, normalization of the vectors, basic weights initialisation, Sammon's error calculation), and the other part of computation (weight renewal, the output of the net calculations) was fulfilled by two threads working in parallel. So, the iterative process of neural network training is parallelised. Each of two processors used in the algorithm performs the training of a part of neural network during each iteration.

While training a neural network,  $d$ -dimensional vectors are presented to the network in pairs  $(X_i, X_j)$ ,  $i, j = 1, \dots, n$ ,  $i \neq j$ , where  $n$  is the whole set of vectors. In the parallel algorithm, two vectors are distributed to different processors and the outputs of neural network for these two vectors are calculated simultaneously. Then the network weights are updated using the error back-propagation algorithm. This process is also performed in parallel. Each processor recalculates the weights of its part of neural network. Data are synchronized after calculating the network outputs for two vectors and after updating the weights of each network layer. A neural network is trained using a fixed number of iterations (one iteration is a part of the training process when all the different pairs are presented to the network once).

The usage of the parallel algorithm is not effective with a small neural network or with low-dimensional datasets because of the data synchronization process.

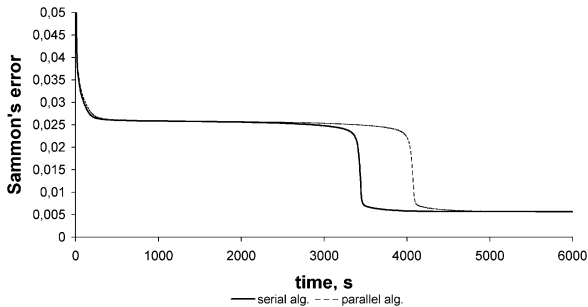
To evaluate the efficiency of the new algorithm two datasets were chosen:

1. Iris Dataset ( $n = 150$ ,  $d = 4$ ) [6].
2. Sonar Dataset [4]. This is the data set used by Gorman and Sejnowski in their study on the classification of sonar signals using a neural network. A real dataset with 208 60-dimensional vectors ( $n = 208$ ,  $d = 60$ ). There are two classes: 111 patterns obtained by bouncing sonar signals off a metal cylinder at various angles and under various conditions and 97 patterns obtained from rocks under similar conditions.

In order to evaluate the algorithm, the feedforward artificial neural network with one hidden layer and two outputs ( $m = 2$ ) was used. In each case, the set of initial weights was fixed in advance. To visualize the Iris dataset, the following parameters were employed: the number of neurons of the hidden layer  $n_2 = 20$ , the learning rate  $\eta = 0.5$  and the momentum value of 0.05. When working with the Sonar dataset, the amount of neurons in the hidden layer was changed:  $n_2 = 500$ .

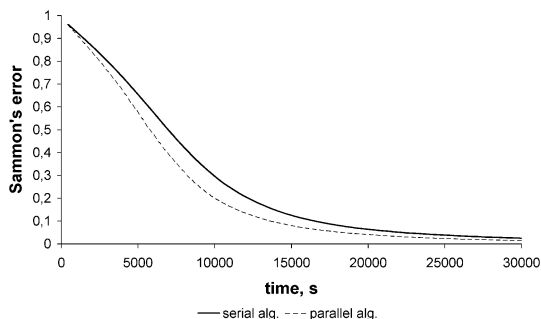
The results of experiments with the parallel algorithm were compared with the serial algorithm results. The mapping error and programme working time have been measured. The experiments to evaluate the second strategy of parallel realization of the SAMANN algorithm were done using different hardware. So, the computation time for the Iris dataset is different as compared with the first strategy. The Dual Xeon SMP system was used to perform the experiments. The result for the Iris dataset is presented in Fig. 6. While working with the Iris dataset, the parallel algorithm execution time is longer than that of the serial algorithm. As it has been said before, this effect occurs because of data synchronization. Using low-dimensional datasets with a small number of hidden neurons, the amount of computations in each neural network learning step is rather small and the expenses of data synchronization are high.

Working with the Sonar dataset, different conclusions may be drawn. The parallel algorithm performs faster than the serial one. Using a 60-dimensional dataset and 500 hidden neurons, we have a large enough amount of computations



**Fig. 6.** Dependence of the projection error on the computation time for the Iris dataset using the second strategy

in each neural network learning step, so the efficiency of the parallel algorithm increases. The result for the Sonar dataset is presented in Fig. 7. Thus, the parallel algorithm performs  $\sim 20\%$  faster than the serial one. By increasing the dimension of the dataset or the amount of neurons in the hidden layer, the efficiency of the parallel algorithm also increases.



**Fig. 7.** Dependence of the projection error on the computation time for the Sonar dataset using the second strategy

## 4 Conclusions

The paper is devoted to the search of the ways of minimizing the SAMANN neural network training time using parallel computations. Two strategies of the SAMANN parallel realization have been proposed and examined. The first strategy, where the training data management is parallelized, does not give the desirable results. That is why the modified algorithm was created. Using the improved strategy, it was managed to better the visualization results. Ideas of the modification of the first strategy may lead to the development of other, more effective strategies.

The second strategy of parallel SAMANN algorithm realization, where the neural network is divided into some parts and the parallel training process of those parts is organized, allows us to train the neural network faster, while working with high-dimensional datasets using rather large neural network. The usage of the proposed parallel algorithm with low-dimensional datasets is not efficient. Working with low-dimensional datasets it is enough to use the serial neural network training algorithm, because the result is obtained rather quickly. While working with high-dimensional datasets, it is important to have a way to speed up the network training process. The proposed parallel algorithm allows us to realize this requirement. The computation time decreases.

The proposed strategies are the attempts to speed up the SAMANN training. Further investigations may lead to new solutions in this area.

## References

1. Blake, C.L., Hettich, S., and Merz, C.J.: UCI repository of machine learning databases. Irvine, CA. University of California, Department of Information and Computer Science (1998). <http://www.ics.uci.edu/~mllearn/MLRepository.html>
2. Borg, I., Groenen, P.: *Modern Multidimensional Scaling: Theory and Applications*, Springer, New York (1997)
3. Ciegis, R.: *Parallel algorithms and net technologies*, Technika, Vilnius (2005) (in Lithuanian)
4. Gorman, R. P., and Sejnowski, T. J.: Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets, in *Neural Networks*, Vol. 1 (1988) 75–89
5. Grama, A., Gupta, A., Karypis, G., Kumar, V.: *Introduction to Parallel Computing*, 2nd Ed. Addison Wesley (2003)
6. Fisher, R. A.: The use of multiple measurements in taxonomic problem. *Annual Eugenics*, Vol. 7, Part II (1936) 179–188
7. Jain, A.K. and Dubes, R.C.: *Algorithms for Clustering Data*. Prentice-Hall (1988)
8. Jain, A.K. and Mao, J.: Artificial neural network for nonlinear projection of multivariate data. *Proc. IEEE International Joint Conference Neural Network*, Vol. 3 (1992) 335–340
9. Jain, A.K., Duin, R., and Mao, J.: Statistical pattern recognition. A review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1 (2000) 4–37
10. Mao, J. and Jain, A.K.: Artificial neural networks for feature extraction and multivariate data projection, *IEEE Trans. Neural Networks*, Vol. 6 (1995) 296–317
11. Medvedev, V., Dzemyda, G.: Optimization of the local search in the training for SAMANN neural network, *Journal of Global Optimization*, Springer, Vol. 35 (2006) 607–623
12. OpenMP C and C++ Application Program Interface Version 2.0, March 2002, OpenMP Architecture Review Board (2002)
13. de Ridder, D., Duin, R.P.W.: Sammon's mapping using neural networks: A comparison. *Pattern Recognition Letters*, Vol. 18 (1997) 1307–1316
14. Sammon, J.J.: A nonlinear mapping for data structure analysis. *IEEE Trans. Computer*, C-18(5) (1969) 401–409
15. Scott, R., Clark, T., Bagheri, B.: *Scientific Parallel Computing*, Princeton University Press (2005)
16. Quinn, M.J.: *Parallel Programming in C with MPI and OpenMP*, McGraw-Hill Inc., New York (2004)

# A POD-Based Center Selection for RBF Neural Network in Time Series Prediction Problems

Wenbo Zhang<sup>1</sup>, Xinchen Guo<sup>2,3</sup>, Chaoyong Wang<sup>2,4</sup>, and Chunguo Wu<sup>2</sup>

<sup>1</sup> College of Computer Science, Jilin Normal University, Siping 136000, China

<sup>2</sup> College of Computer Science and Technology, Jilin University, Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, Changchun 130012, China

<sup>3</sup> College of Science, Northeast Dianli University, Jilin 132012, China

<sup>4</sup> Department of Fundamental Sciences, Jilin Teacher's Institute of Engineering and Technology, Changchun 130021, China  
wucg@jlu.edu.cn

**Abstract.** Center selection based on proper orthogonal decomposition (POD) is presented to select centers for the radial basis function (RBF) neural network in prediction of nonlinear time series. The proposed method takes advantages of the time-sequence feature in time series data and enables the center selection to be implemented in a parallel manner. Simulations on a benchmark problem and on two predictions of stock prices show that the presented method can be applied effectively to the prediction of nonlinear time series. Besides possessing higher precisions in training and testing, the proposed method has stronger generalization and noise resistance abilities, compared to several other popular center selection methods.

## 1 Introduction

Radial basis function (RBF) neural network is a feed forward three-layered network. With the excellent abilities of global optimal solution search and function approximation, it has been studied and applied widely, especially, in the fields of nonlinear system identification and prediction [1~3]. In RBF neural networks, the weights from input to hidden units are fixed to be 1. Each hidden unit has a basis function. All basis functions transform input data nonlinearly and map them into a high dimensional space, which is named feature space. The output units perform linear combinations of the data from hidden units. The radial basis function used most usually is Gauss function, which can be formulated as:

$$\varphi(x) = \exp(-\|x - c\|^2 / (2\sigma^2)) \quad (1)$$

where  $c$  is called the center, and  $\sigma$  the band width. The general information of radial basis functions and other alternative forms can be found in Ref. [4]. In general, for Gauss basis functions both the centers and band widths are chosen differently with each other. Researchers have noticed that the center selection is a key factor to obtain better results when applying RBF neural network to concrete problems [4, 5].

The main classical center selection approaches include random method, hard c-means (HCM) method [6, 7], Kohonen method [8], nearest neighbor clustering method[9], adaptive fuzzy c-means method[10, 11] and some others. The random method is too rough, which could result in big difference in each running time. When a certain class is empty, it is difficult to adjust the empty class by using the HCM method and Kohonen method. Furthermore, when there are new samples added to the training set, most of existing approaches need to repeat the selection process from scratch. In order to overcome the deficiencies mentioned above, we propose a method of center selection for time series prediction RBF neural network based on proper orthogonal decomposition (POD). This proposed method takes advantages of time-sequence relationship inhering in time series data and keeps the orders among adjacent data. Therefore, it is more reasonable for time series prediction problems. When a certain class is empty, what one should do to adjust this class is only to reduce the segment number. Moreover, this proposed method supports parallel computing and online training. The center selection is performed in sub-series and all centers can be extracted at the same time. New centers can be extracted from the newly obtained samples and the centers corresponding to the earliest sub-series can be removed easily to keep the center number fixed. To examine the validity of the proposed method, we compare it with several other popular center selection methods.

## 2 HCM and OLS Center Selections

### 2.1 HCM Center Selection

As a simple and effective method of unsupervised learning, HCM has been used extensively to categorize data [12]. In the construction of RBF neural networks it can be used to determine the centers from a training set  $S = \{x_i | x_i \in R^n, i = 1, 2, \dots, N\}$ . Suppose the number of classes to be clustered is  $n_r$ . The implementation of HCM can be summarized as:

1. Let  $k = 0$  and initialize a sorting matrix  $U^0$  randomly,

$$u_{ij} = \begin{cases} 1 & x_i \in A_j \\ 0 & x_i \notin A_j \end{cases} \quad (2)$$

where  $A_j$  stands for the  $j$ th class ( $i=1, 2, \dots, N, j = 1, 2, \dots, n_r$ ).

2. Calculate centers  $c_j$ ,

$$c_j = \frac{1}{\sum_{i=1}^N u_{ij}} \sum_{i=1}^N u_{ij} x_i \quad (j = 1, 2, \dots, n_r) \quad (3)$$

3. Compute the new sorting matrix  $U^{k+1}$ ,

$$u_{ij} = \begin{cases} 1 & d_{ij} = \min_{1 \leq l \leq n_r} \{d_{il}\} \\ 0 & \text{else} \end{cases} \quad (4)$$

where  $d_{ij} = \|x_i - c_j\|$  ( $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, n_r$ ) are the Euclidian distance between  $i$ th sample and  $j$ th center.

4. If the Frobenius norm difference between the latest two sorting matrix is smaller than  $\varepsilon$  (a predetermined small positive constant), i.e.,

$$\|U^{k+1} - U^k\|_F \leq \varepsilon$$

then terminate the procedure and output the class centers,  $c_i$  ( $i=1,2,\dots,n_r$ ), or else, let  $k=k+1$  and return to step 2.

The final outputs of vectors are used as the RBF centers in the HCM-based RBF neural network and the weights are trained by using the least square iteration.

### 2.2 OLS Center Selection

OLS algorithm was first applied to select the centers of RBF neural networks by Chen et al. (1991) in Ref. [13]. It has been widely accepted as an effective method for center selection in RBF network [13, 14]. The OLS method [15] treats the process of determining a RBF neural network as a regression problem. All centers and weights (from hidden neurons to output neuron) are decided simultaneously in OLS method. At the beginning the OLS method takes all samples as centers. Thus the regression problem has the following form

$$d(t) = \sum_{j=1}^N p_j(t)\theta_j + e(t) \tag{5}$$

where  $p_j(t) = \exp(-\|x(t) - c_j\|^2 / (2\sigma^2))$ ,  $p_j(t)$  is the regression factor,  $e(t)$  is the error term between the expected and actual outputs. After discretizing the time variable  $t$  into  $N$  points, and changing Eq. (5) into a matrix form, we have

$$d = P\theta + e \tag{6}$$

where  $d = [d(1), d(2), \dots, d(N)]$ ,  $P = [p_1, p_2, \dots, p_N]$ ,  $p_j = [p_j(1), p_j(2), \dots, p_j(N)]^T$ ,  $e = [e(1), e(2), \dots, e(N)]^T$ ,  $\theta = [\theta_1, \theta_2, \dots, \theta_N]^T$ .

Note  $P$  is an  $N \times N$  square matrix.  $P$  can be decomposed into the below form

$$P = WA \tag{7}$$

where  $W$  is an orthogonal matrix, and  $A$  an upper triangle matrix. The columns of  $W$  can be taken as a group of bases in a certain space. Then Eq. (6) can be transformed into

$$d = Wg + e \tag{8}$$

where  $g = A\theta$ . Due to the least square method, we can get the estimation of  $g$ , whose components are

$$\hat{g}_j = W_j^T d / W_j^T W_j, (j = 1, 2, \dots, N) \tag{9}$$



From Eq. (9) we can define the error rate for all bases  $w_j$  in a descent way

$$\tilde{\epsilon}_j = \hat{g}_j^2 w_j^T w_j / d^T d, (j = 1, 2, \dots, N) \quad (10)$$

which give the contribution rate of  $w_j$  to the variance of  $d$ . The OLS method selects bases according the following rule

$$1 - \sum_{j=1}^{n_r} \tilde{\epsilon}_j < \rho \quad (11)$$

where  $\rho$  is a predetermined constant, named tolerance. The bases corresponding to the first  $n_r$  largest  $\tilde{\epsilon}_j$  are selected, and the samples corresponding these bases are taken as centers to construct the RBF neural network [16]. OLS can provide the centers related to the most significant contribution to approximation error reduction.

### 3 Methodology

The proper orthogonal decomposition (POD) is a powerful and elegant method for data analysis aimed at obtaining low-dimensional approximate descriptions of a high-dimensional process [17, 18]. It is an important and essential technique for data reduction and feature extraction, and has been widely used in various disciplines including image processing, signal analysis, data compression, process identification, and especially in many other engineering fields, such as adaptive control, distributed reacting system, nondestructive detection, and some others [19~21]. In general, there are two different interpretations for the POD. The first interpretation regards the POD as the Karhunen-Loève decomposition (KLD) and the second one considers that the POD consists of three methods: the KLD, the principal component analysis (PCA), and the singular value decomposition (SVD). Because of the close connections and the equivalence of the three methods, the authors prefer the second interpretation, that is, the POD includes the KLD, PCA and SVD. To investigate the equivalence between the three methods, Refs. [22] and [23] can be referred.

In recent years, POD has been widely investigated as a powerful tool for model reduction and usually used to extract the dominant coherent structures from a system. These dominant coherent structures are utilized to realize the model reduction combined with Galerkin method [24]. However, it has been scarcely used to select centers in construction of RBF neural networks. In this paper the action of nonlinear time series is treated as a dynamical system and the prediction issue as model reduction under a given precision. Then the POD method is utilized to the prediction of nonlinear time series.

Let  $L$  observations of a nonlinear time series be  $\{x_l | x_l \in R, l = 1, 2, \dots, L\}$ . Note that here  $x_l$  ( $l = 1, 2, \dots, L$ ) are scalars. Now the goal is to predict the next observation after each  $n$ -observation window. Then the training set can be denoted as  $S = \{x_i | x_i \in R^n, i = 1, 2, \dots, N, N = L - n\}$ , where the last sample is eliminated due to its lack of target. Let the RBF neural network have  $n_r$  centers. The proposed method of

center selection is to divide the  $L$ -observation sequence into  $n_r$  segments, where it is supposed that  $L$  can be divided exactly by  $n_r$ . Each segment has  $m$  observations ( $m = L/n_r$ ). For example, the first segment consists of observation data from  $x_1$  to  $x_m$ , and the second segment consists of observation data from  $x_{m+1}$  to  $x_{2m}$ , ....., the last segment (*i.e.* the  $n_r$  th segment) consists of observation data from  $x_{L-m+1}$  to  $x_L$ . Table 1 shows the segment division in the whole observation sequence.  $K$  ( $K = m-n+1$ ) samples can be extracted from the  $j$ th ( $j=1,2,\dots,n_r$ ) segment by using window-sliding method. And then the samples constitute a  $K \times n$  matrix  $X_j$ , whose  $k$ th row is corresponding to the  $k$ th sample belonging to the  $j$ th segment. The main idea of the proposed algorithm is to perform POD on the  $n$ -order square matrices  $X_j^T X_j$  ( $j=1,2,\dots,n_r$ ) and extract the first proper orthogonal basis for each segment. These extracted proper orthogonal bases would be taken as centers in the construction of RBF neural networks. So after performing POD on all segments, the desired  $n_r$  centers could be obtained.

**Table 1.** Segment division in the L-observation sequence

Segments	Data distribution in each segment
1st	$x_1, x_2, \dots, x_{m-1}, x_m$
2nd	$x_{m+1}, x_{m+2}, \dots, x_{2m-1}, x_{2m}$
.....	.....
$n_r$ th	$x_{L-m+1}, x_{L-m+2}, \dots, x_L$

The proposed method can be described as follows:

(1) Set the number of centers  $n_r$ .

(2) Organize samples in the  $j$ th segment into a  $K \times n$  matrix  $X_j$  ( $j=1,2,\dots,n_r$ ) and calculate the first proper orthogonal basis of  $X_j^T X_j$ . And then normalize it and denote the normalized first proper orthogonal basis as  $c_j$ , which is taken as the  $j$ th center( $j=1,2,\dots,n_r$ ).

(3) Calculate the sorting matrix  $U$  whose elements are

$$u_{ij} = \begin{cases} 1 & d_{ij} = \min_{1 \leq l \leq n_r} \{d_{il}\} \\ 0 & \text{else} \end{cases} \tag{12}$$

where  $d_{ij} = \|x_i - c_j\|$  ( $i=1,2,\dots,N, j=1,2,\dots,n_r$ ) are the Euclidian distance between the  $i$ th sample and  $j$ th center.

(4) Calculate the center widths  $\sigma_j$

$$\sigma_j = \frac{1}{\sum_{i=1}^N u_{ij}} \sum_{i=1}^N u_{ij} d_{ij}, \quad (j = 1, 2, \dots, n_r) \tag{13}$$

(5) Output all centers  $c_j$  derived from first proper orthogonal basis of all segments.

## 4 Numerical Experiments

This paper proposed a novel view point to handle the time series prediction problems. To evaluate the validity of the proposed method, three data sets are considered, one well-known benchmark data set and two real application data sets.

The benchmark data set is derived from Mackey-Glass (MG) system. The MG system was constructed by Mackey and Glass in 1977 for a model of blood cell regulation and became quite common as artificial forecasting benchmark. The high chaotic property of MG system makes it challenging for researchers in the field of time series prediction. Consequently, it is studied widely in time series prediction communities. The MG system can be formulated as follows [25~27]:

$$\frac{dx}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t) \quad (14)$$

where  $a = 0.2$ ,  $b = 0.1$ ,  $\tau = 17$ ,  $\Delta t = 0.1$  (for numerical integration),  $t \in (0, 800)$ . Let  $x(t) = 1.0$ , for any  $t \leq 0$ , then 8 thousands data points are obtained.

The first real application data set is obtained from Shanghai Stock Exchange (SHSE), consisting of 353 data points from the 4th January 1999 to the 30th June 2000. The first 153 data are taken as training samples and the rest as testing samples. And the second real application data set is obtained from New York Stock Exchange (NYSE), consisting of 153 data points from the 6th January 2001 to the 10th June 2001. For the two real application data sets, the first is used to train and test the RBF neural network, and the second to evaluate the generalization and noise resistance abilities of the network trained from the first real application data set.

In this paper one-step prediction is adopted. Experiences show that data regularization is very important to the model accuracy. Hence, the regularization is used in our experiments according the following rule:

$$x_l = \frac{x_l}{\max_{1 \leq k \leq L} (x_k)}, \quad (l = 1, 2, \dots, L) \quad (15)$$

The embedding dimensions of Santa Fe Time Series Competition Data Set D and MG system are taken as 9 and 6, respectively, as what were chosen in most published literatures [25, 26]. Because usually a Stock Exchange works five days in each week, we attempt to take the embedding dimensions of both real data sets as 3, to make it larger than the half of the number of consecutive transaction days. And the numbers of hidden units in this paper are obtained by pruning method. At the beginning a relative larger number of hidden units is set, and then calculate the centers and the sample number of each center. If there are centers which have samples less than two, the number of hidden units is reduced by 1, and then re-calculate the centers and the sample number of each center until all centers have samples no less than two. In this paper the number of hidden units for benchmark problem is 15, and that for real application data sets are both 5.

For HCM-based and POD-based RBF neural networks, the weights are decided by using least square iteration method. The parameters  $\eta$ ,  $\theta$  and  $\rho$  are taken as 0.003, 0.0004 and 0.001, respectively, and the maximal step number in the least square

iteration method as 2000. The training precision, testing precision, generalization ability and noise resistance are compared among three RBF neural networks based on different center selection methods, including POD, HCM and OLS approaches. Table 3 shows the errors for different approaches. The formula to compute the errors is as follows

$$e = \frac{1}{2N} \sum_{i=1}^N \frac{(y_i - \hat{y}_i)^2}{y_i^2} \tag{16}$$

where  $y_i$  and  $\hat{y}_i$  represent the actual target value and that predicted by RBF neural networks for sample  $x_i$ , respectively. To measure the improvement of the proposed method, the improved rates are defined as follows

$$\mu = \frac{e_{HCM} - e_{POD}}{e_{HCM}} \tag{17}$$

$$\nu = \frac{e_{OLS} - e_{POD}}{e_{OLS}} \tag{18}$$

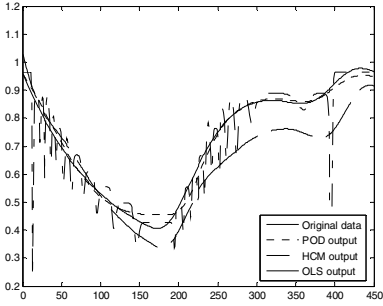
where  $e_{POD}$ ,  $e_{HCM}$  and  $e_{OLS}$  are the errors corresponding to the methods of POD, HCM and OLS, respectively. Tables 2 and 3 show the training and testing results for the benchmark and real application data sets, respectively. It can be seen from these tables that the proposed center selection method improves the prediction accuracy drastically and the improved rates range from 51% to 92%, compared with HCM and OLS based methods. It is reported in Ref. [27] that the conventional neural networks results in a training RMSE 0.024 for the MG system. Compared with the reported neural network, the proposed method improves the accuracy around by 25%, resulting a training RMSE 0.018. Figs. 1~7 also give the comparison results, where the solid line represents actual data, the dotted line for prediction based on POD neural network, the dash-dot line for prediction based on HCM neural network and dashed line for prediction based on OLS neural network. To view it clearly, we solely present the result for the proposed method in Fig. 3 for the benchmark data set. All of the seven figures demonstrate that the HCM-based method performs worst, the OLS-based method performs better and the proposed method performs best in training, testing, generalization and noise resistance aspects.

**Table 2.** Comparison errors (MSE) for benchmark data set

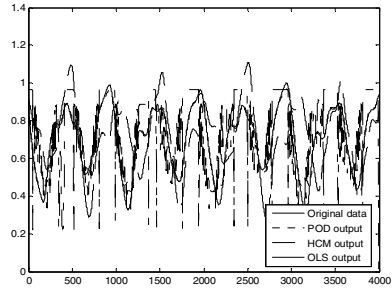
	HCM	OLS	POD	Improved Rate	
				$\mu$	$\nu$
Training (MG system)	0.00290	0.00053	0.00033	89%	61%
Testing (MG system)	0.00603	0.00108	0.00053	91%	51%

**Table 3.** Comparison errors (MSE) for real data sets

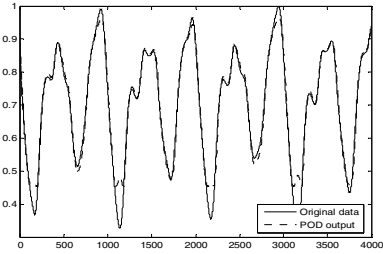
	HCM	OLS	POD	Improved Rate	
				$\mu$	$\nu$
Training (SHSE)	0.00316	0.00084	0.00036	88%	57%
Testing (SHSE)	0.00540	0.00098	0.00042	92%	57%
Generalization (NYSE)	0.00069	0.00141	0.00012	82%	91%
Noise resistance (NYSE)	0.00409	0.00174	0.00066	83%	62%



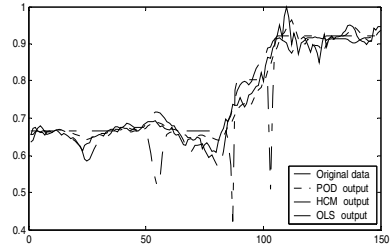
**Fig. 1.** Comparison of training precision for benchmark dataset (MG System)



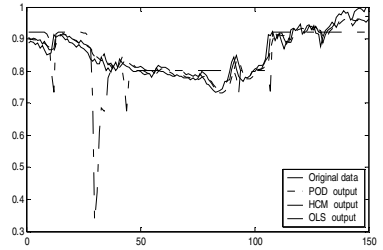
**Fig. 2.** Comparison of testing precision for benchmark dataset (MG System)



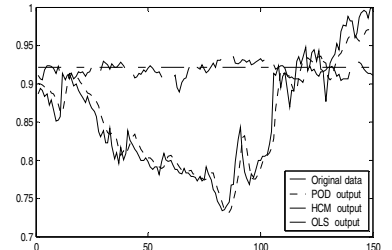
**Fig. 3.** Training precision for benchmark dataset (MG System)



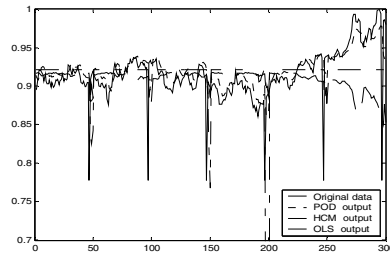
**Fig. 4.** Comparison for training precision (SHSE)



**Fig. 5.** Comparison for testing precision (SHSE)



**Fig. 6.** Comparison for generalization ability (NYSE)



**Fig. 7.** Comparison for noise resistance (NYSE)

## 5 Conclusions and Discussions

Proper orthogonal decomposition (POD) is introduced to select centers for the radial basis function (RBF) neural network for the prediction of nonlinear time series. The nonlinear time series is treated as a dynamical system and the prediction issue is regarded as a model reduction under a given precision. The proposed method could capture more dynamic features of the time series data, since it takes advantage of the time-sequence relationship among time series data. The experiments regarding to real stock price predictions and benchmark problem demonstrate the proposed method is effective and has high training and testing precisions as well as high generalization and noise resistance abilities.

## References

1. Chi W., Zhou B., Shi A.G., Cai F., Zhang Y.S.: Radial basis function network for chaos series prediction. *Lecture Notes in Computer Science*. 3174 (2004) 920-924
2. Sheta A.F., De Jong K.: Time-series forecasting using GA-tuned radial basis functions. *Information Sciences*. 133 (2001) 221-228
3. Rivas V.M., Merelo J.J., Castillo P.A., Arenas M.G., Castellano J.G.: Evolving RBF neural networks for time-series forecasting with EvRBF. *Information Sciences*. 165 (2004) 207-220
4. Lee D.W., Lee J.: A novel three-phase algorithm for RBF neural network center selection. *Lecture Notes in Computer Science* 3173 (2004) 350-355
5. Zhu M.X., Zhang D.L.: RBF neural network center selection based on Fisher ratio class separability measure. *IEEE Transactions on Neural Networks*. 13 (5) (2002) 1211-1217
6. Miyamoto S.: Information clustering based on fuzzy multi-sets. *Information Processing & Management*, 39(2) (2003) 195-213
7. Meng L., Wu Q.H., Yong Z.Z.: A genetic hard c-means clustering algorithm. *Dynamics of Continuous Discrete and Impulsive Systems-Series B-Applications & Algorithms* 9 (3) (2002) 421-438
8. Tarkov M.S., Mun Y., Choi J.Y., Choi H.I.: Mapping adaptive fuzzy Kohonen clustering network onto distributed image processing system. *Parallel Computing* 28 (9) (2002) 1239-1256
9. Chan P.T., Rad A.B.: Adaptation and learning of a fuzzy system by nearest neighbor clustering. *Fuzzy Sets and Systems* 126 (3) (2002) 353-366

10. Fan J.L., Zhen W.Z., Xie W.X.: Suppressed fuzzy C-means clustering algorithm. *Pattern Recognition Letters*. 24 (9-10) (2003) 1607-1612
11. Wu K.L., Yang M.S.: Alternative c-means clustering algorithms. *Pattern Recognition* 35 (10) (2002) 2267-2278
12. Oh S.K., Pedrycz W., Park H.S.: Multi-FNN identification based on HCM clustering and evolutionary fuzzy granulation. *Simulation Modeling Practice and Theory* 11 (2003) 627-642
13. Chen S., Cowan C.F., Grant P.M.: Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Transactions on Neural Networks*, 2 (2) (1991) 302-309
14. Chen S., Member S., Wu Y., Luk B. L.: Combined Genetic Algorithm Optimization and Regularized Orthogonal Least Squares Learning for Radial Basis Function Networks. *IEEE Transactions on Neural Networks*. 10 (5) (1999) 1239-1243
15. Wang X.X., Brown D.J.: Boosting orthogonal least squares regression. *Lecture Notes in Computer Science* 3177 (2004) 678-683
16. Abido M.A., Abdel-Magid Y.L.: Adaptive tuning of power system stabilizers using radial basis function networks. *Electric Power Systems Research*. 49 (1999) 21-29
17. Chatterjee A.: An introduction to the proper orthogonal decomposition. *Current Science*. 78 (7) (2000) 808-816
18. Holmes P., Lumley J.L., Berkooz G.: *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, Cambridge (1996)
19. Kunisch K., Volkwen S.: Control of the Burgers Equation by a Reduced-Order Approach Using Proper Orthogonal Decomposition. *Journal of Optimization Theory and Applications*. 102 (2) (1999) 345-371
20. Shvartsman S.Y., Theodoropoulos C., Rico-Martinez R., Kevrekidis I.G., Titi E.S., Mountziaris T.J.: Order reduction for nonlinear dynamic models of distributed reacting systems. *Journal of Process Control* 10 (2000) 177-184
21. Banks H.T., Joyner M.L., Wincheski B., Winfree W.P.: Nondestructive evaluation using a reduced-order computational methodology. *Inverse Problems* 16 (2000) 929-945
22. Liang Y.C., Lee H.P., Lim S.P., Lin W.Z., Lee K.H., Wu C.G.: Proper orthogonal decomposition and its application—Part I: theory. *Journal of Sound and Vibration*. 252 (2002) 527-544
23. Wu C.G., Liang Y.C., Lin W.Z., Lee H.P., Lim S.P.: A note on equivalence of proper orthogonal decomposition methods. *Journal of Sound and Vibration* 265 (2003) 1103-1110
24. Azeez M.F.A., Vakakis A.F.: Proper orthogonal decomposition (POD) of a class of vibroimpact oscillations. *Journal of Sound and Vibration*. 240 (5) (2001) 859-889
25. Müller K.R., Smola A.J., Rätsch G., Schölkopf B., Kohlmorgen J., Vapnik V.N.: Using Support Vector Machines for Times Series Prediction. In B. Schölkopf, C. Burges and A. Smola (Eds), *Advances in Kernel Methods Support Vector Learning*, MIT Press (1998): 243-253
26. Flake G. W., Lawrence S.: Efficient SVM regression training with SMO. *Machine Learning*. 46 (2002) 271-290.
27. Maguire L.P., Roche B., McGinnity T.M., McDaid L.J., Predicting a chaotic time series using a fuzzy neural network, *Information Sciences* 112 (1998) 125-136

# Support, Relevance and Spectral Learning for Time Series

Bernardete Ribeiro

Department of Informatics Engineering, Center for Informatics and Systems,  
University of Coimbra, Polo II, P-3030-290 Coimbra, Portugal  
bribeiro@dei.uc.pt

**Abstract.** This paper proposes the Spectral Clustering Kernel Machine (SCKM) for times series prediction. Support Vector Machine (SVM), Relevance Vector Machine (RVM) and the Spectral Clustering Kernel Machine (SCKM) are compared in terms of performance accuracy for a simple time series approximation problem. The three outlined algorithms each of which with interesting features to perform automated learning are examined, analysed and empirically tested. In case of the SVM, our tests combine also a preprocessing stage including Kohonen Maps (SOM) as well as K-means clustering. In the case of RVM we also implemented a constructive approach based on the fast marginal likelihood maximization described in [14]. Prediction results in two benchmark time series have been addressed using various performance metrics. The results demonstrate that whereas RVM models achieve larger parsimony of the fitted model, both SVM and SCKM attain higher accuracy. The learning models are competitive for real world problems.

## 1 Introduction

Time series forecasting is a challenge in many fields such as engineering, business, medicine and many other application domains. Many techniques exist for the approximation of the underlying process of a time series: linear methods such as ARX, ARMA, etc. [7] and non linear approaches such as Support Vector Machines (SVM) [9]. The SVM technique [15] has strong theoretical foundation [3] and has strong practical potential. An alternative learning machine amenable to probabilistic interpretation for regression is the relevance vector machine (RVM) [13]. In [11] a relevance vector machine with adaptive kernels for time series prediction is presented. Both approaches have shown practical relevance not only for classification and regression problems but also, more recently, in unsupervised learning. In particular, RVM provides an attractive framework for Bayesian learning of sparse kernel regression models. Bayesian methods allow for the incorporation of prior information allowing the user to make coherent inference which automatically embodies Occam's razor quantitatively [8]. The strength characteristics of both methods has been addressed in [5] which presents a Bayesian formulation of SVM for regression.

In general, these methods try to build a model of the process. The model is then used on the last values of the series to predict the future values. The



common difficulty to all the methods is the determination of sufficient and necessary information for an accurate prediction. In recently years Spectral Clustering [10] algorithm has proven successful in separating non-convex groups of data as shown by a number of applications on real world problems. The method finds structure in data using spectral properties of a pairwise matrix, therefore it appears as a simple yet powerful algorithm for time-series forecasting.

This paper proposes the Spectral Clustering Kernel Machine (SCKM) for time series prediction. The method of spectral clustering is compared with support vector learning, sparse Bayesian learning for the Box-Jenkins and Mackey-Glass benchmarks in terms of performance accuracy. The paper is organized as follows. Section 2 reviews shortly Support Vector Machines (SVM). In Sect. 3 we introduce briefly the Relevance Vector Machines (RVM). A constructive algorithm is also addressed. In Sect. 4 Spectral Clustering is introduced. Section 5 presents the data sets and discusses the results. Finally the conclusions and future work are addressed in Sect. 6.

## 2 Support Vector Machines

Support Vector Machines (SVM) belong to the family of powerful kernel-based learning machines. SVM combine essentially two strong ideas: maximum margin classifiers with low capacity and implicit features spaces defined by kernel functions [15]. In other words, they conjoint the following properties: low Vapnik-Chervonenkis (VC) dimension solutions through maximization of the margin and kernel nonlinearity. These properties bring good generalization to the Vapnik learning machine. Given a training data set consisting of input-output pairs  $\{\mathbf{x}_n, t_n\}_{n=1}^N$  SVM use the convolution of the scalar product to build, in input space, the nonlinear decision functions of the form:

$$f(\mathbf{x}) = \sum_{n=1}^N w_n K(\mathbf{x}, \mathbf{x}_n) + w_0 \quad (1)$$

$K$  represents the kernel (or mapping) function, a positive semi-definite matrix:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)\dot{\phi}(\mathbf{x}_j) = \exp(-\beta\|\mathbf{x}_i - \mathbf{x}_j\|^2) \quad (2)$$

$\phi$  is mapping function from input space to feature space and weights  $w$  are given by the non-zero Lagrange multipliers called support vectors (SVs).

## 3 Relevance Vector Machines

Relevance Vector Machines (RVM) are a probabilistic non-linear model with a prior on the weights that promotes sparse solutions. Given a set of training samples  $\{\mathbf{x}_n\}_{n=1}^N \in \mathbb{R}^d$  and output targets  $\{t_n\}_{n=1}^N \in \mathbb{R}$ , the outputs are a linear combination of the response of a set of  $M$  basis functions  $\phi(\mathbf{x}) = \{\phi_j(\mathbf{x})\}_{j=1}^M$ :

$$y_i = f(\mathbf{x}_i) = \sum_{j=1}^M w_j \phi_j(\mathbf{x}_i) + w_0 = \mathbf{w}^T \phi(\mathbf{x}_i) \quad , \quad (3)$$

where  $w_0$  is the bias in the regression model,  $\phi_j(\mathbf{x}_i)$  is the output of the  $j$ th basis function to input sample  $\mathbf{x}_i$  and the following vectors are defined:  $\mathbf{y} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^T$ ,  $\mathbf{w} = [\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_M]^T$ ,  $\phi(\mathbf{x}_i) = [1, \phi_1(\mathbf{x}_i), \dots, \phi_M(\mathbf{x}_i)]^T$  and  $\Phi(\mathbf{x})_{ij}$  is the  $N \times (M + 1)$  design matrix.

Inferring weights distribution is an ill-posed problem due to risk of overfitting, thus regularization is required. A Gaussian prior distribution of zero mean and variance  $\sigma^2 \equiv \alpha_j^{-1}$  is then defined over each weight:

$$p(\mathbf{w}|\alpha) = \prod_{j=1}^M \mathcal{N}(w_j|0, \alpha_j^{-1}) = \prod_{j=1}^M \sqrt{\frac{\alpha_j}{2\pi}} \exp\left\{-\frac{1}{2}\alpha_j \mathbf{w}_j^2\right\}. \quad (4)$$

RVM carries out estimation of the hyperparameters and weights. In this procedure some of the hyperparameters grow to infinity thus causing the corresponding weights to shrink towards zero. This means that the training vectors  $\mathbf{x}_n$  with the corresponding hyperparameter  $\alpha_n$  above a certain threshold are pruned out from the model (3). Thus, only the vectors with weights different zero, called relevant vectors (RV), will be used.

### 3.1 Constructive Approach: Fast Marginal Likelihood Maximization

The constructive approach of the RVM is described in [14]. RVM performs local maximization with respect to  $\alpha$  of the marginal likelihood:

$$\mathcal{L}(\alpha) = \log p(\mathbf{t}|\alpha, \sigma^2) = -\frac{1}{2}[M \log 2\pi + \log |\mathbf{C}| + \mathbf{t}^T \mathbf{C}^{-1} \mathbf{t}].$$

For convenience of the algorithm description  $s_i$  and  $q_i$  are defined by  $s_i \triangleq \phi(\mathbf{x}_i)^T \mathbf{C}_{-i}^{-1} \phi(\mathbf{x}_i)$  and  $q_i \triangleq \phi(\mathbf{x}_i)^T \mathbf{C}_{-i}^{-1}$  where  $\mathbf{C}_{-i}$  represents the covariance matrix  $\mathbf{C}$  with  $i$ th basis function removed. This covariance matrix is defined by:

$$\mathbf{C}_{-i} = \mathbf{C} - \alpha_i \phi_i \phi_i^T.$$

Then the algorithm proceeds as follows: If  $q_i^2 > s_i$   $\mathcal{L}(\alpha)$  will have its local maximization so the vector  $\mathbf{x}_i$  will be added to the model as a relevance vector and  $\alpha_i$  is updated according to  $\alpha_i = \frac{s_i^2}{q_i^2 - s_i}$ , otherwise maximization will be reached at infinity and  $\alpha_i = \infty$ .

## 4 Spectral Clustering Kernel Machine

The concept initially inspiring spectral clustering is graph partitioning [4]. A generalization to arbitrary number of clusters is found in [10] following the work in [16]. The algorithm typically involves constructing an affinity matrix  $A$  from the data, taking an eigen-decomposition of this matrix, and then applying traditional clustering techniques, such as K-means, to a subspace of the eigenvectors.

<sup>1</sup>  $\mathbf{C} = \sigma^2 \mathbf{I} + \Phi \mathbf{A}^{-1} \Phi^T$  where  $\mathbf{A} = \text{diag}(\alpha_i), i = 1, \dots, M$ .

This subspace is found by specifying that there are  $K$  clusters, and thus using the first  $K$  eigenvectors as the clustering space.

In essence the algorithm groups  $N$  data points into a number of predefined  $K$  clusters. It constructs an affinity matrix from the data which is a  $N \times N$  matrix whose elements  $A(i, j) = e^{-d(x_i, x_j)/2\sigma^2}$  are the pairwise similarities of the points. This is done using a kernel with  $\sigma$  being the kernel width and  $d(x_i, x_j) = \|x_i - x_j\|^2$ . However, other affinity definitions are possible. The affinity matrix is then normalised to form a matrix  $L = D^{-1/2}AD^{-1/2}$  where  $D = \text{diag}(\sum_{j=1}^d A_{ij})$ . As pointed out in [12] the matrix  $D$  takes into account the spread of the various clusters, i.e., points belonging to less dense clusters get lower sums in the corresponding rows of  $A$ .  $L$  is positive definite with eigenvalues smaller or equal to 1. The first  $K$  eigenvectors are then computed and arranged as columns in a matrix  $\hat{Y}$ . The rows of  $\hat{Y}$  are then normalised and treated as  $K$ -dimensional vectors; performing  $K$ -Means on these vectors will return the desired clustering.

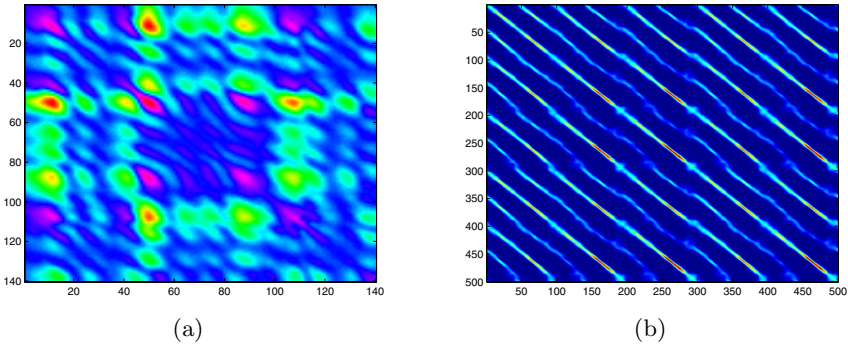


Fig. 1. Affinity matrix (a) Box Jenkins (b) Mackey-Glass

## 5 Results and Discussion

In time series prediction, the previous values and the current value are used as inputs for the prediction model. One-step ahead prediction is referred as Short-Term Prediction whereas when multi-step ahead predictions are needed, it is called a Long-Term Prediction problem. The two benchmarks time series chosen fall into each of one of these types.

### 5.1 Gas Furnace of Box Jenkins

The Box and Jenkins Furnace (BJF) data are from [2]. There are originally 296 data points  $\{y(t), u(t)\}$ , from  $t = 1$  to  $t = 296$ .  $y(t)$  is the output  $CO_2$  concentration and  $u(t)$  is the input gas flow rate. Here we predict  $y(t)$  based on 10 parameters  $\{y(t-1), y(t-2), y(t-3), y(t-4), u(t-1), u(t-2), u(t-3), u(t-4)$ ,

**Table 1.** Box Jenkins Time Data Series

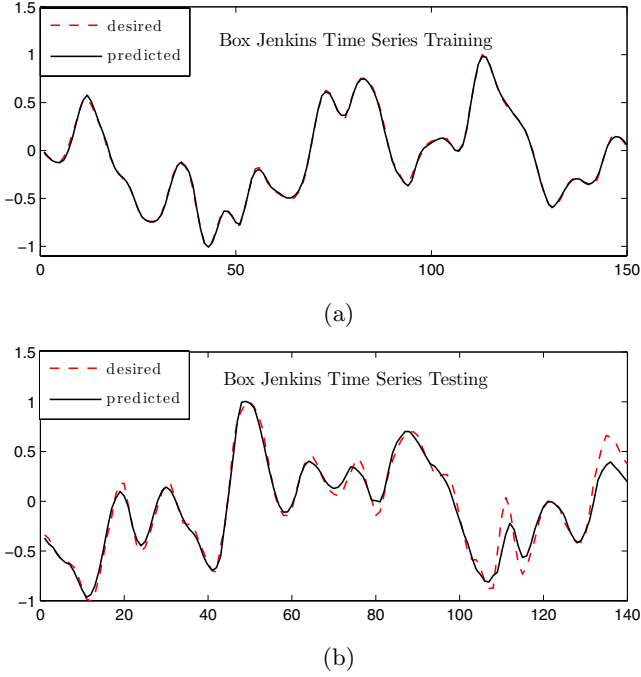
	EVs	RMS1	SCC1	RMS2	SCC2
SVM	79	0.0176	0.9993	0.0956	0.9802
SOM + SVM	24	0.0727	0.9767	0.0830	0.9701
K-MEANS + SVM	33	0.0291	0.9962	0.1774	0.9828
RVM	14	0.0520	0.9976	0.1045	0.9942
RVM-F	10	0.0829	0.9816	0.1662	0.9622
SCKM	10	0.054	0.9934	0.044	0.9949

**Table 2.** Mackey-Glass Time Data Series

	EVs	RMS1	SCC1	RMS2	SCC2
SVM (P=6)	90	0.062	0.9912	0.065	0.9927
SVM (P=84)	111	0.063	0.9858	0.064	0.9863
RVM (P=6)	98	0.005	0.9999	0.025	0.9995
RVM (P=84)	82	0.028	0.9996	0.049	0.9981
RVM (P=6, noise)	41	0.051	0.9987	0.067	0.9978
RVM (P=84, noise)	59	0.066	0.9978	0.076	0.9958
RVM-F (P=6)	84	0.034	0.9995	0.049	0.9989
RVM-F (P=6, noise)	76	0.054	0.9985	0.066	0.9978
SCKM (P=6, noise)	20	0.027	0.9930	0.028	0.9935

$u(t-5), u(t-6)$ . This reduces the number of effective data points to 290. Although most methods find that the best set of input variables for predicting  $y(t)$  is  $\{y(t-1), u(t-4)\}$  we have used all the 10 parameters on the available data sets: the first column is the output variable  $y(t)$ , the remaining columns are the input variables  $\{y(t-1), y(t-2), \dots, u(t-6)\}$ . We have splitted the data into a 150 training set size and a 140 test set size.

Table 1 illustrates several learning machines for the Box Jenkins problem. We applied (i) SVM, (ii) a Self-Organising Map (SOM) prior to SVM, and (iii) a K-means Clustering algorithm combined with an SVM. We run the time series under the RVM framework described in the (iv) standard Tipping version [13]. We implemented a (v) constructive approach based on the fast marginal likelihood maximization algorithm, Relevance Vector Machine Fast (RVM-F) accordingly to previously described. Finally, a Spectral Clustering Kernel Machine (SCKM) was set up to test results obtained for forecasting the time series against previous approaches. The number of Support Vectors (SVs) in SVM, the number of Relevance Vectors (RVs) in RVM and the number of EigenCluster Centers (ECs) are also indicated under the column header designated by Expansion Vectors (EVs) in the Table 1. The values of the Root Mean Square (RMS) error and Squared Correlation Coefficient (SCC) are also therein indicated where index 1 denotes training samples whereas index 2 denotes test samples. Clearly, K-means, acting as a pre-processing stage, improves SVM results (SCC2=0.9828). However, this is not verified when applying the SOM algorithm which might indicate



**Fig. 2.** SVM: Box Jenkins Time Series for (a) training and (b) testing

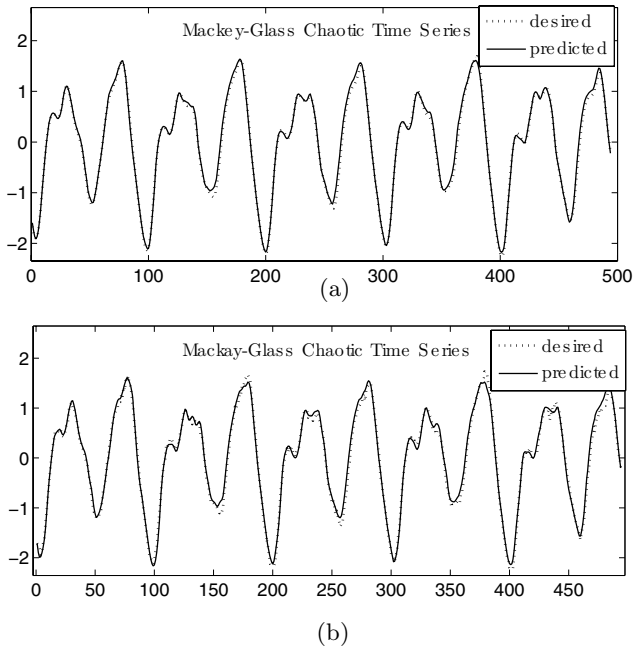
us that clustering of information plays a major role than its mapping organization. It is interesting to notice that the baseline version of RVM (SCC2=99.42) presents a better result when compared with the RVM-F machine which gets slightly worst performance (SCC2=96.22). This might be due to the fact that with the constructive approach the sparsity of the sought solution is fully explored allowing a larger economy of kernels in the final model. We observe only 10 kernel functions (RVM-F) as compared with 5 kernel functions (baseline RVM) since in the former, the  $\alpha_i$  different from zero are incrementally constructed during the iterative maximization procedure. SCKM is the best method since it achieves a good performance result (SCC2=0.9949) in the test data time series. The corresponding affinity matrix is shown in Fig. 1(a).

### 5.2 Mackey-Glass Chaotic Time Series

The Mackey-Glass equation was originally developed for modeling white blood cells production. It is a chaotic time series making an ideal representation of the non linear oscillations of many physiological processes. It is a time delay ordinary differential equation derived by integrating:

$$\dot{x}[t] = \frac{ax[t - \tau]}{1 + x[t - \tau]^{10}} - bx[t] \tag{5}$$

When  $a = 0.2$ ,  $b = 0.1$  and  $\tau = 17$ , the integration produces a chaotic time series [1]. Following the standard approach [6] each training sample contains the points  $x[t-18]$ ,  $x[t-12]$ ,  $x[t-6]$  and  $x[t]$ . Prediction is then made to the future point  $x[t+6]$ . The task is to use currently available points in the chaotic time series to predict future points at  $t+P$  where  $P = 84$ . Figure 3 compares the desired and predicted Mackey-Glass series at  $P = 84$  for both cases without and with noise using RVMs. We observe from Table 2 that the solution found with RVMs for the series forecasting is more economic in terms of number of basis functions than the one obtained by SVMs while preserving the accuracy property shown by the latter. Likewise, SCKM achieves comparable performance with previous approaches.



**Fig. 3.** RVM: Mackey-Glass series at  $P = 84$  without noise (a) and (b) with noise

## 6 Conclusions

In this paper we present a new approach (SCKM) based on the spectral clustering algorithm for tuning of time series. We compare the results of the new approach with the two state-of-the-art kernel learning machines (SVM) and (RVM) for prediction of time series using Box Jenkins and Mackey-Glass benchmarks. Support, relevance and spectral learning lead to machine approaches with different yet competitive properties. In summary, a SVM finds the separating surface, which is expressed as a linear combination of kernel functions centered at support data vectors, by selecting borderline and misclassified examples; a RVM

uses a similar linear combination centered in the relevant vectors but instead describes the separating surface by selecting typical instances or prototypes; in the SCKM approach the algorithm finds the clusters of data in a semi-supervised manner taking advantage of their spectral properties. Examining the results we find out that while RVM finds a more parsimonious prediction model (the number of RVs is, sometimes, one or more orders less than SVM) SVM and SCKM attain higher accuracy. The performance metrics indicate that the three outlined approaches do not produce overfitting as compared with more traditional learning machines. Future work aims to combine the simplicity of the spectral clustering algorithm with the relevance learning of the eigenvectors in order to address real world applications where data is noisy and scarce.

## References

1. E M Azoff. *Neural Networks Time Series Forecasting of Financial Markets*. Wiley, Sussex, UK, 1994.
2. G.E.P. Box and G.M. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden Day, San Francisco, USA, 1970.
3. Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39(1):1–49, 2001.
4. J. Shi J. and J. Malik. Normalized cuts and image segmentation. 22(8):888–905, 2000.
5. James T.-Y. Kwok. The evidence framework applied to support vector machines. *IEEE Transactions on Neural Networks*, 11(5):1162–1173, 2000.
6. James T.Y. Kwok and D.Y. Yeung. Constructive neural networks: Some practical considerations. In *IEEE International Conference on Neural Networks (ICNN'94)*, pages 198–203, Orlando, Florida, USA, June 1994.
7. L. Ljung. *System identification theory for User*. Prentice-Hall, Englewood Cliffs, 1987.
8. David J Mackay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, UK, 2003.
9. K.-R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Artificial Neural Networks — ICANN'97*, pages 999 – 1004, Berlin, 1997. Springer Lecture Notes in Computer Science, Vol. 1327.
10. A. Y. Ng, I. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, Cambridge, MA, 2002.
11. Joaquin Quiñero-Candela and Lars Kai Hansen. Time series prediction based on the relevance vector machine with adaptive kernels. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 985–988, Piscataway, New Jersey, 2002. IEEE.
12. G Sanguinetti, J Laidler, and N D Lawrence. Automatic determination of the number of clusters using spectral algorithms. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 55–60, Connecticut, USA, September 28 - 30 2005.

13. M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, June 2001.
14. M. E. Tipping and Anita Faul. Fast marginal likelihood maximisation for sparse bayesian models. In Christopher M. Bishop and Brendan J. Frey, editors, *International Workshop on Artificial Intelligence and Statistics*, 2003.
15. V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, New York, 1995.
16. Y. Weiss. Segmentation using eigenvectors: a unifying view. Technical report, CS. Dept., UC Berkeley, 1999.



# Support Vector Machine Detection of Peer-to-Peer Traffic in High-Performance Routers with Packet Sampling

Francisco J. González-Castaño<sup>1</sup>, Pedro S. Rodríguez-Hernández<sup>1</sup>,  
Rafael P. Martínez-Álvarez<sup>1</sup>, and Andrés Gómez-Tato<sup>2</sup>

<sup>1</sup> Departamento de Ingeniería Telemática,  
Universidad de Vigo, Spain

ETSI Telecomunicación, Campus, 36310 Vigo, Spain

<sup>2</sup> CESGA, Spain

{javier,pedro,rmartinez}@det.uvigo.es,  
agomez@cesga.es

**Abstract.** In this paper, we explore the possibilities of support vector machines to identify peer-to-peer (p2p) traffic in high-performance routers with packet sampling. Commercial networks limit user access bandwidth -either physically or logically-. However, in research networks there are no *individual* bandwidth restrictions, since this would interfere with research tasks. User behavior in research networks has changed radically with the advent of p2p multimedia file transfers: many users take advantage of the huge bandwidth (e.g. compared to domestic DSL access) to exchange movies and the like. This behavior may have a deep impact on research network utilization. Consequently, in the framework of the MOLDEIP project, we have proposed to apply support vector machine detection to identify those activities in high-performance research network routers. Due to their high port rates, those routers cannot extract the headers of all the packets that traverse them, but only a sample. The results in this paper suggest that support vector machine detection of p2p traffic in high-performance routers with packet sampling is highly successful and outperforms recent approaches like [1].

## 1 Introduction

In this paper, we apply support vector machines [2,3,4] to detect peer-to-peer (p2p) traffic in high-performance routers with packet sampling. Commercial networks limit user access bandwidth -either physically or logically-. However, in research networks there are no *individual* bandwidth restrictions, since this would interfere with research tasks. This is the case of the Galician Network of Science & Technology (RECETGA), which comprises seven campuses and many other research institutions in NW Spain, with over 100,000 users (<http://www.cesga.es/en/defaultE.html>).

User behavior in research networks has changed radically with the advent of p2p multimedia file transfers: many users take advantage of the huge bandwidth (e.g. compared with domestic DSL access) to exchange movies and the like. This

behavior may have a deep impact on research network utilization nowadays. We have identified similar concerns in other research networks ([http://security.uchicago.edu/peer-to-peer/no\\_fileshare.shtml](http://security.uchicago.edu/peer-to-peer/no_fileshare.shtml)).

The increasing usage of p2p software in the last three years has raised the need of p2p detection tools. There are some related initiatives. Among them, in [5], the authors describe a methodology to identify p2p flows at the transport layer, based on connection patterns of p2p flows, regardless of packet payloads. The authors point that these patterns are more difficult to conceal than explicit flow-conveyed information. In [6], the authors propose to detect p2p traffic by identifying protocol-dependent signatures (key strings) in TCP packet payloads. This is unfeasible in high-performance routers that *sample* packet headers. However, it may be valid for low-to-medium performance routers. In [1], the authors present a Bayesian classifier of protocols. Although the mean classification accuracy (across all application types) is quite good, the results for p2p traffic are quite poor (an accuracy of 56%).

The MOLDEIP p2p activity detection tool we have developed (*i*) is independent from router performance (*ii*) is transparent to the users and (*iii*) works with *sampled* packet headers<sup>1</sup>.

MOLDEIP does not consider individual flows but the average activity of individual IP addresses. Consequently, there is no short-term technological dependence (first goal). Recent network activity is taken from *netflow export files* [7], and thus we do not consider packet payloads (unlike the approach in [6]). As in [5], the system is transparent to network users because it does not scan their machines (second goal). It works with packet headers, and it does not check them all, but only a sample (third goal).

Off-line analysis is feasible because p2p traffic is a nuisance, but it does not disable the network. Thus, a 24-hour margin to take corrective actions is acceptable.

MOLDEIP relies on a support vector machine (SVM). As far as we know, this is an original approach. Reference [8] proposes SVMs to detect anomalous traffic. However, it focuses on intrusion attacks instead of p2p traffic identification. The results in this paper suggest that support vector machine detection of p2p traffic in high-performance routers with packet sampling is highly successful and outperforms recent approaches like [1].

Specifically, we consider the problem of constructing SVM classifiers based on a given classification of  $m$  training vectors (*points*) in the  $n$ -dimensional space  $\mathbb{R}^n$ , represented by the  $m \times n$  matrix  $D$ , given the membership of each IP point  $D_i$ ,  $i = 1, \dots, m$  in one of two classes - “innocent” or “guilty”. Each point  $D_i \in \mathbb{R}^n$  is a vector representing an IP address within RECETGA’s ranges, whose components indicate particular aspects of the behavior of that IP address.

For this problem, we follow the linear programming model in [9]:

$$\begin{aligned} & \min_{w,z,\gamma,y} e'y + e'z \\ & \text{such that } C(Dw - e\gamma) + \frac{1}{\alpha}y \geq e \\ & \quad -z \leq w \leq z, \quad y \geq 0, \quad z \geq 0, \end{aligned} \tag{1}$$

---

<sup>1</sup> MOLDEIP project, Xunta de Galicia grant PGIDIT03TIC00101CT.

where  $w$  is a vector of separator coefficients,  $y$  is a vector of slack variables,  $\gamma$  is an offset,  $\alpha$  is an error penalty,  $C$  is a  $m \times m$  diagonal matrix with plus ones or minus ones depending on the class of the points represented by  $D$  rows, and  $e$  stands for a vector of ones of appropriate dimension. This model allows us to employ state-of-the-art linear programming solvers. If we compare (I) with the original quadratic model in [2], we see that it tries to minimize the 1-norm of  $w$  parameters instead of their 2-norm (the 2-norm maximizes the margin  $\frac{2}{w'w}$  between the bounding planes  $x'w = \gamma \pm 1$  in the standard formulation).

We briefly comment our notation: Capital Latin letters are sets or matrices depending on the context. Lower case Latin letters denote vectors in  $\mathbb{R}^n$ , except for the range  $i, \dots, q$  that denotes integers. Lower case Greek letters are real scalars. Subindices are different components, i.e.,  $x_i$  is the  $i$ -th component of the  $n$ -component vector  $x$  and  $a'b$  is the inner product  $\sum_{i=1}^n a_i b_i$ . For any vector, a  $K$  subindex denotes a subvector whose components have indices belonging to the set  $K$ . For any matrix  $B$ ,  $B_i$  is its  $i$ -th row, and  $B_K$  is the submatrix composed of all  $B_i$  such that  $i \in K$ . For any entity, a superindex is an iteration index.

We can define the following desirable quality goals:

- Obviously, a high classification accuracy.
- As few nonzero components in  $w$  as possible, for speeding the classifier up and identifying statistically relevant components.
- As few active constraints at the solution of problem (I) as possible. The corresponding set  $D_K$  of *support vectors* represents problem (I), i.e. the solution does not change if we drop the remaining vectors. When *updating* classifiers, it is interesting to add to the next problem as few representatives of previous training datasets as possible -i.e. their support vector sets-.

The rest of this paper is organized as follows: in Sect. 2 we model p2p detection. In Sect. 3 we evaluate detection with *full* monitoring of packet headers. In Sect. 4 we evaluate detection from *sampled* packet headers, the main goal of this work. In Sect. 5 we compare our approach with previous methods. Finally, Sect. 6 concludes the paper.

## 2 Problem Modeling

We define a new problem each 24-hour slot, from a netflow data batch compiled in the previous slot.

At the time this paper was written, all RECETGA network traffic of the Vigo Campus traversed *both* a CISCO 7206 router and a Juniper M10 router. The former produced binary netflow files comprising the headers of *all* traversing packets (origin and destination IP addresses, port identifiers, protocol identifiers, etc), whereas the latter only extracted the information of 0.1% packets.

First, we convert the netflow files to ASCII format with *Flow-tools* [10]. The resulting 24-hour files approximately occupy 2 GB for the CISCO 7206 and 50

MB for the Juniper M10. Each line in these files corresponds to a single end-to-end transfer, with the following fields: *unix time*, *number of transferred packets*, *transfer size* in bytes, *origin IP address*, *destination IP address*, *origin port*, *destination port* and *transport protocol* (TCP or UDP).

Note that there is no flow nor session information: for a given IP address within RECETGA range, there may be thousands of end-to-end IP transfers that appear in the ASCII file as independent entries.

Basically, our preprocessor generates a dataset with *a single entry* per RECETGA IP address.

**Remark 1:** The fields in each dataset entry match the current high-level metrics in RECETGA graphic analysis tools.

Previous analyses [11] have identified the block sizes and packet formats of most popular p2p protocols nowadays. However, We do not consider explicit p2p protocol information (which can be concealed or encrypted). Our parameters comprise different aggregation levels (day time, night time, 5-minute time slots and 1-hour time slots), to consider temporal behavior.

Let  $x$  be a RECETGA IP address. Its dataset entry has the following fields:

```

p1: number of different IP addresses  $x$  sets connections to, day time
p2: number of different IP addresses  $x$  admits connections from, day time
p3: number of different ports in  $x$  external IP addresses access, day time
p4: number of different ports  $x$  opens, day time
p5: total traffic  $x$  generates, day time (MB)
p6: total traffic  $x$  receives, day time (MB)
p7 – p12: same as p1-p6, night time
p13: average number of different IP addresses  $x$  sets connections to, day time, measured in
      5-minute slots
p14: same as p13, standard deviation
p15: same as p13, maximum
p16: average number of different IP addresses  $x$  admits connections from, day time, meas. in
      5-minute slots
p17: same as p16, standard deviation
p18: same as p16, maximum
p19: average number of different ports in  $x$  external IP addresses access, day time, meas. in
      5-minute slots
p20: same as p19, standard deviation
p21: same as p19, maximum
p22: average number of different ports  $x$  opens, day time, meas. in 5-minute slots
p23: same as p22, standard deviation
p24: same as p22, maximum
p25: average traffic  $x$  generates in a 5-minute slot, day time (MB)
p26: same as p25, standard deviation
p27: same as p25, maximum
p28: average traffic  $x$  receives in a 5-minute slot, day time (MB)
p29: same as p28, standard deviation
p30: same as p28, maximum
p31 – p48: same as p13-p30, night time
p49 – p66: same as p13-p30, 1-hour measurement slots
p67 – p84: same as p49-p66, night time

```

### 3 Performance Evaluation, CISCO 7206

We selected a representative netflow file for the evaluations in this section, corresponding to all RECETGA traffic March 28 2003. This file contained a typical

RECETGA behavior, clearly including background p2p traffic, without unusual events like attacks or network failures.

**Remark 2:** As previously said, at the time this paper was written, all Vigo Campus traffic traversed both the CISCO 7206 and the Juniper M10. Since the CISCO 7206 monitored all traffic, we analyzed its logs (*i*) to determine the effectiveness of SVMs for p2p off-line detection and (*ii*) to determine statistically relevant parameter classes. The main goal of this work is p2p detection from *sampled* data, as imposed by the Juniper M10. We study that case in Sect. 4.

### 3.1 Training and Testing Sets

First, we applied two filters to the problem:

1. The preprocessor only generated entries for IP addresses belonging to *Universidad de Vigo*: 6141 IP addresses passed this filter.
2. The preprocessor only generated entries for IPs exchanging over 1000 KB/day (**threshold** parameter): 614 IP addresses passed both filters.

The preprocessing times on a Pentium IV (2.4 GHz, 512 MB DDR) are the following: 17.5 hours for p1–p12, 1.25 hours for p13–p48 and 7.25 hours for p49–p84. There are three types of parameters: *IP parameters*, such as p1 and p2, *port parameters*, such as p3 and p4 and *traffic parameters*, such as p5 and p6. We observe that 12-hour IP and port parameters are clearly dominant in preprocessing time. Obtaining the number of *different* IP addresses and ports satisfying a given condition for a given IP address involves scanning the whole input file and creating a new entry in a temporary list each time a *new* IP address or port is found (which implies scanning the whole list that far). Thus, setting long slots (12 hours for p1–p12) implies long temporary lists to explore per new-entry check, and long preprocessing times as a consequence.

Once we obtained the dataset, we labeled it as follows:

1. “*Guilty*” entries: those satisfying any of the following conditions:
  - (a) using TCP ports of well-known P2P protocols.
  - (b) large night downloads:  $p_{12} > 100 \times \text{threshold}$
  - (c) relatively large uploads:  $((p_5 > 10 \times \text{threshold}) \text{ AND } (p_5 > p_6)) \text{ OR } ((p_{11} > 10 \times \text{threshold}) \text{ AND } (p_{11} > p_{12}))$
2. “*Innocent*” entries: otherwise.

Thus we obtain an initial collection of “known points” to train the classifier, although it is possible to feed SVM training tools with unlabeled points [12].

**Remark 3:** RECETGA labeling criteria may vary with network regulations.

**Remark 4:** for further validation of labeling criteria 1-2, we checked the IP addresses in the dataset that established connections with well-known p2p servers. We detected 29 such addresses, which we had correctly labeled as “guilty” ones.

456 “innocent” and 158 “guilty” IP addresses resulted. We denote the percentage of “guilty” points as  $p$  (in this case,  $p = 0.26$ ). Finally, we divided the labeled

dataset into one hundred random partitions of *training* and *testing* subsets (90% and 10% points, respectively). To compensate class asymmetry (typical in real datasets [13]), rather than applying a unique factor  $1/\alpha$  to all error variables in (II), we weight them differently depending on point class:  $1/p\alpha$  for “innocent” points and  $1/(1-p)\alpha$  for “guilty” ones.

### 3.2 Parameter Options in SVM Training

We solved problem (II) on each training subset, and then tested the resulting classifier on the corresponding testing subset. Table I shows average results (across the 100 experiments) for parameter options **A** and **B**, as explained below (*note*: we tuned  $\alpha$  in (II) in preliminary trials;  $N_w$  is the average number of nonzero classifier coefficients at the solution;  $N_{sv}$  is the average number of support vectors at the solution;  $a_g$  and  $a_i$  are average blind testing accuracies on the testing subsets for “guilty” and “innocent” IP addresses, respectively).

**A**: all parameters participate in the model.

**B**: only day-time port parameters participate. This choice is motivated by three facts: first, malicious users may decide to hide behind the bulk of day-time traffic. Second, the parameters that depend on the number of IP addresses will be unfeasible with the advent of the huge IPv6 addressing range. Third, the size of “normal” data transfers keeps growing with the capacity of servers and links. For example, soon all p2p movie exchanges will consist of full DVD contents. Thus, traffic parameters have a strong short-term technological dependence. Note that the labeling criteria in Sect. 3.1 are useless as *detection* criteria with option B.

**Table 1.** Average results of SVM trainer (II), options A and B,  $\alpha = 10$

Option	$N_w$	$N_{sv}$	$a_g$	$a_i$
A	44	25%	74.1%	92.9%
B	11	42.4%	65.9%	93.4%

We observe some interesting facts:

- The testing accuracy for “guilty” points is low:  $\sim 65\%$  for option B.
- The testing accuracy for “innocent” points is high, as good as 93.4% for option B. However, this is less important for RECETGA managers than the performance for “guilty” points.
- As it could be expected, the number of points supporting the classifier ( $N_{sv}$ ) grows when limiting the number of parameters. In any case, even with option B it is possible to discard 60% of them.
- The number of statistically significant parameters is surprisingly low for option B, for a similar performance.

Option B seems the most interesting in practice: it avoids malicious user concealment to a great extent, provides a fast classifier (few parameters) and supports the problem with  $\sim 40\%$  points on average. However, in principle, it is not valid due to its low testing accuracy for “guilty” points. In the following section we propose a strategy to correct this problem.

### 3.3 $\gamma$ Shift

Rather than applying more complex tools to improve the testing accuracy for “guilty” points, we propose a simple strategy to balance  $a_g$  and  $a_i$ . “Guilty” points tend to form a “small” cloud at the edge of the “innocent” region in parameter space. Consequently, once the classifier is available, we “tune” it by shifting  $\gamma$ .

We group the 100 training scenarios into 10 super-scenarios, each one consisting of a testing set with 10 scenarios and a training set with 90 scenarios. The testing sets of any two super-scenarios do not overlap.

Then, for a given super-scenario, we take the best classifier (according to the testing accuracies in the previous section) across all 90 scenarios of the training set and we tune its  $\gamma$  value until, on average, the accuracies for “innocent” and “guilty” points are similar across all 90 scenarios of the training set. The testing accuracy of a super-scenario is the average accuracy of its tuned classifier across all 10 scenarios in its testing set.

The resulting average accuracies of the tuned classifiers are well balanced. For option A,  $a_g=90.2\%$  and  $a_i=87.1\%$ . For option B,  $a_g=79\%$  and  $a_i=78.7\%$ . We observe that the testing accuracy for “guilty” points grows to  $\sim 80\%$  for option B, which is much better than the results in [1].

For RECETGA managers, a high detection accuracy for “guilty” points is extremely important. In other networks, avoiding false positives may be preferable. Note that  $\gamma$ -shift is a parameterizable process that can be adjusted accordingly.

### 3.4 Faster Preprocessing Stage

Solving (1) with CPLEX took few seconds in all tests, so preprocessing is dominant in solution time. From the experience of RECETGA managers, port parameters are significant. Thus, in order to decrease preprocessing time, we defined two new parameter options:

- C:** day-time port parameters, aggregation levels of 1 hour and 5 minutes.
- D:** day-time port parameters, aggregation level of 5 minutes.

Table 2 shows the performance of trainer (1) with  $\gamma$  shift for these options. We observe that, although the average number of support vectors grows slightly, restricting parameters to 5-minute aggregation also yields the following:

1. Although all parameters in option D are significant, there are only 6 such parameters: p19-p24. In the training scenarios, these parameters appear frequently at the solution with nonzero weights: p19 in 100% scenarios, p20 in 99%, p21 in 99%, p22 in 87%, p23 in 87% and p24 in 100%.

**Table 2.** Results of SVM trainer (II) with  $\gamma$  shift, options B-D,  $\alpha = 10$ 

Option	$N_w$	$N_{sv}$	$a_g$	$a_i$
B	11	42.4%	79%	78.7%
C	10	44.3%	78.3%	78.1%
D	6	47.2%	81.6%	81.4%

- An average testing accuracy  $a_g$  of 81.6%. In other words, only 2-3 guilty IP addresses escape in each scenario, on average. Apparently, a 5-minute aggregation level captures the evolution of user behavior along the day.

If we preprocess the netflow files with the parameter options so far, the following preprocessing times result: 26 hours for option A, 7.75 hours for option B and 3.25 hours for option C. For option D, we only need  $\sim 34$  min of Pentium IV CPU to preprocess 24 hours of RECETGA monitoring data. Parallel computing may drastically reduce this preprocessing time.

## 4 Performance Evaluation, Juniper M10

In this section we evaluate the impact of packet sampling on p2p detection accuracy. The Juniper M10 (M10 in the sequel) cannot store all traversing packet headers. In RECETGA, the M10 samples 0.1% of them. As a result, we relaxed M10 labeling, by setting `threshold` to 1 KB/day.

**Remark 5:** this obvious change is the only design decision to admit sampled data, i.e. we use exactly the same programs and methodology in Sect. 3, but we feed them with the sampled headers of the M10.

Table 3 shows SVM detection performance March 28 2003, when RECETGA traffic traversed *both* the CISCO 7206 and the M10. The preprocessing stage was much faster on M10 sampled data: full parameter preprocessing took less than 30 seconds. So there is no advantage, in terms of preprocessing time, in defining different aggregation levels.

Note that the methodology in Sect. 3 is even better on M10 data. The results for option D are specially remarkable: *for less significant port parameters (3 versus 6), and a comparable average number of support vectors, the average testing accuracy for “guilty” points is practically 90% after  $\gamma$  shift*. In all training scenarios, the three significant parameters in option D are p19, p21 and p24.

**Table 3.** Average results of SVM trainer (II). Accuracies after  $\gamma$  shift. Options B-D, CISCO 7206 and Juniper M10 datasets, March 28 2003. C: CISCO, J: Juniper,  $\alpha = 10$ .

Option	$N_w$ C/J	$N_{sv}$ C/J	$a_g$ C/J	$a_i$ C/J
B	11/5	42.4%/36.6%	79%/87.8%	78.7%/87.5%
C	10/5	44.3%/38.9%	78.3%/90.8%	78.1%/88.7%
D	6/3	47.2%/50.3%	81.6%/89.3%	81.4%/83.2%



Next we evaluated the performance of SVM p2p detection along seven days. Table 4 shows the result. We observe that the testing accuracies for “guilty” points did not change significantly in a year.

Finally, we evaluated SVM classifier “persistence”. We applied the July 15 2004 classifier to detect p2p traffic the following four business days, i.e, without re-training the classifier each day. Table 5 shows the results.

We observe that the testing accuracies are quite persistent. There may be a slight performance decrease when using a classifier across different days, although we meet RECETGA goals in all cases (over 80% success). Probably, it is enough to recompute classifiers twice a week.

**Table 4.** Average results of SVM trainer (II). Accuracies after  $\gamma$  shift. Options B-D, Juniper M10 datasets, July 15-21 2004,  $\alpha = 10$ .

Option	$N_w$	$\sigma_{N_w}$	$N_{sv}$	$\sigma_{N_{sv}}$	$\bar{a}_g$	$\sigma_{a_g}$	$\bar{a}_i$	$\sigma_{a_i}$
B	6	0.83	36.7%	7.1%	89.4%	5.07%	88.9%	5.46%
C	5	0.79	40.4%	7.5%	89.0%	4.22%	88.4%	4.7%
D	2	0.88	60.3%	17.1%	87.6%	4.99%	83.1%	6.46%

**Table 5.** Persistence of the SVM classifier of July 15 2004, when applied the next four business days. Accuracies after  $\gamma$  shift. Options B-D, Juniper M10 datasets.

Option	$\bar{a}_g$	$\sigma_{a_g}$	$\bar{a}_i$	$\sigma_{a_i}$
B	90.27%	5.35%	91.37%	5.35%
C	88.68%	2.09%	90.05%	1.05%
D	83.9%	3.47%	82.95%	1.88%

## 5 Comparison with Other Methods

The solution in [6] relies on packet payloads, which are not available in our scenario. Our approach outperforms [1], which has a 55.18% success at best.

The approach in [5] focuses on suspicious flows whereas ours focuses on suspicious IPs (RECETGA managers are interested in detecting *machines*). However, as line speed grows, it becomes impossible to work at flow level. Indeed, *our methodology is impervious to sampling*, whereas the first heuristic in [5] (“source-destination IP pairs that concurrently use both TCP and UDP”) would certainly fail on a sample with 0.1% packets: although large TCP flows would still be detected, most UDP traffic would not.

Moreover, our approach easily adapts itself to new p2p protocols and user practices: it only demands a periodic training stage to identify the parameters that are currently relevant. So, it is harder to set countermeasures, since it is not based on predefined heuristics.

Finally, the criteria in [5] is not acceptable in our context, since it is based on payload data analysis. This is forbidden by RECETGA’s rules.

## 6 Conclusions

High performance routers cannot monitor all the packet headers, due to their high line rates. As a consequence, techniques such as [5] are unfeasible.

Off-line support vector machine detection of sampled p2p traffic is highly accurate. A few day-time port activity parameters measured in 5-minute slots (option D) suffice for this purpose. This is interesting because neither traffic or IP parameters -currently short-term technologically dependent- are strictly necessary. The aggregation level in option D reflects measurement data variability, yielding detection accuracies of  $\sim 90\%$  for “guilty” points (M10, Table 3). The  $\gamma$ -shift postprocessing we propose has been a key tool to achieve these results, which are highly competitive with [1].

An unexpected bonus of packet sampling is a dramatic decrease in preprocessing time, from several minutes (for option D) to seconds (regardless of the parameter selection). Thus, it is possible to preprocess a 24-hour netflow file and generate the corresponding classifier in less than a minute. This suggests that, even for routers capable to monitor all traversing packets, it may be wise to discard most packets (as many as 99,9%) before training the classifier.

## References

1. Moore, A. W., Zuev, D.: Internet Traffic Classification Using Bayesian Analysis Techniques. In Proc. ACM Sigmetrics (2005)
2. Vapnik, V. N.: The nature of statistical learning theory. Springer, New York, (1995)
3. Burges, C. J. C.: A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2** (1998) 121–167
4. Cristianini, N., Shawe-Taylor J.: An introduction to support vector machines. Cambridge University Press, (2000)
5. Karagiannis, T., Broido, A., Faloutsos, M., Claffy, K.C.: Transport Layer Identification of P2P Traffic. In Proc. ACM IMC (2004)
6. Sen, S., Spatscheck, O., Wang, D.: Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures. In Proc. 13th International Conference on World Wide Web. (2004)
7. NetFlow:  
<http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml>.
8. Tran, Q.A., Duan, H., Li, X.: One-Class Support Vector Machine for Anomaly Network Traffic Detection. 2nd Network Research Workshop. 18th APAN. (2004)
9. Bradley, P.S., Mangasarian, O. L., Musicant, D. R.: Optimization Methods in Massive Datasets. In *Handbook of Massive Datasets*. Abello, J., Pardalos, P. M., Resende, M. G. C. Editors, Kluwer Publishing, (2002) 439–472
10. Fullmer, M.: Flow-tools. <http://www.splintered.net/sw/flow-tools/>
11. Karagiannis, T., Broido, A., Brownlee, N., Claffy, K.C., Faloutsos, M.: File-sharing in the Internet: A characterization of P2P traffic in the backbone. Technical report. Department of Computer Science University of California. Riverside. <http://www.cs.ucr.edu/~tkarag/papers/tech.pdf> (2003)
12. Fung, G., Mangasarian, O. L.: Semi-Supervised Support Vector Machines for Unlabeled Data Classification. Data Mining Institute Technical Report. 1999 99-05. October 1999. *Optimization Methods and Software*. **15** (2001) 29–44
13. Murphy, P. M., Aha, D.W.: UCI repository of machine learning databases. <ftp.ics.uci.edu/pub> (1992)

# Improving SVM Performance Using a Linear Combination of Kernels

Laura Dioşan<sup>1,2</sup>, Mihai Oltean<sup>1</sup>, Alexandrina Rogozan<sup>2</sup>,  
and Jean-Pierre Pecuchet<sup>2</sup>

<sup>1</sup> Computer Science Department, Babeş-Bolyai University,  
Cluj-Napoca, Romania

{lauras, moltean}@cs.ubbcluj.ro

<sup>2</sup> LITIS, Institut National des Sciences Appliquées,  
Rouen, France

{arogozan, pecuchet}@insa-rouen.fr

**Abstract.** Standard kernel-based classifiers use only a single kernel, but the real-world applications and the recent developments of various kernel methods have emphasized the need to consider a combination of multiple kernels. We propose an evolutionary approach for finding the optimal weights of a combined kernel used by the Support Vector Machines (SVM) algorithm for solving some particular problems. We use a genetic algorithm (GA) for evolving these weights. The numerical experiments show that the evolved combined kernels (ECKs) perform better than the convex combined kernels (CCKs) for several classification problems.

## 1 Introduction

Various classification techniques have been used for assigning the correct labels associated to each input item. Kernel-based techniques (such as Support Vector Machines (SVMs) [1]) are one of the intensively explored classifiers used for solving this problem. Standard kernel-based classifiers use only a single kernel, but the real-world applications and the recent developments of various kernel methods have emphasized the need to consider a combination of kernels. If this combination is a linear one, than one has to determine the weights associated to each kernel.

The optimal values for these weights could be determined with different mathematical methods (such as Sequential Minimal Optimisation (SMO) [2], Semidefinite Programming [3]). Recently, the evolutionary methods (such as Evolutionary Algorithms (EAs) [4]) were used as an alternative approach for solving various optimization problems. EAs can solve most problems without taking into account the constraints regarding the continuity and the derivability of the functions that encode the core of the problems. Also, the evolutionary methods can be faster than the convex techniques in some situations (for problems with many variables or many instances). The convex technique can solve very well these problems, but the running time is larger than that of evolutionary approach. Even if the evolutionary techniques are able to find in some cases

only an approximation of solution, this approximation is quickly found, compared to the convex tools. It is a trade-off between the solution accuracy and the computing time.

Therefore, EAs could be used for finding the optimal weights of a combined kernel. In our research we have used Genetic Algorithms (GAs) [5] for evolving the kernel weights in order to obtain a more complex kernel. A real encoding is used for representing the GA chromosomes. Each gene of a chromosome is associated to a kernel weight. For computing the fitness of a chromosome, we embed the complex kernel into a SVM algorithm and we run this algorithm for a particular classification problem. The accuracy rate computed by the SVM algorithm (on a validation set) represents the quality of the current chromosome. The accuracy rate (on the test dataset) computed by the SVM algorithm, which uses the evolved kernel, is compared to those computed by a single-kernel SVM algorithm for several classification problems. Numerical experiments show that an ensemble of multiple kernels is always superior to individual kernels, for the considered problems, in terms of classification accuracy (the number of correctly classified items over the total number of items). The obtained results also show the ability of the GAs to evolve better weights for the combination of kernels than the convex method (proposed in [3], [6]).

The paper is organized as follows: Sect. 2 describes some related work in the field of combined kernel generation. Section 3 describes the proposed technique for evolving combined kernels. This is followed by a special section (Sect. 4) where the results of the experiments are presented. Finally, Sect. 5 concludes.

## 2 Related Work

Only two attempts for finding the weights of a combined kernel for SVM algorithm were found in the literature. Recently, Lanckriet et al. [3] and Sonnenburg [6] considered the conic combinations of kernel matrices for the SVM and showed that the optimization of the coefficients of such a combination reduces to a convex optimization problem known as a quadratically constrained quadratic program. They shown how the kernel matrix can be learnt from data via semi-definite programming [3] and via semi-infinite linear programming [6]. The other attempt is also made by Lanckriet [7] and it is an improved approach of [3]. He proposed a novel dual formulation of the QCQP as a second-order cone-programming problem.

Both approaches are matrix-based techniques because, in both cases, the algorithms learn the kernel matrix (or the Gram matrix) associated to the multiple kernel. The model selection can be viewing in terms of Gram matrices rather than kernel functions.

Our model can be considered an evolutionary alternative to these approaches. We will call, in what follows, the learnt kernel of Lanckriet and Sonnenburg *convex combined kernel (CCK)* – because it is found using a convex method – and we will call our kernel *evolved combined kernel (ECK)* – because it is learnt using evolutionary techniques.

### 3 Proposed Model

#### 3.1 Representation

Standard SVM algorithm works with a particular kernel function, which is empirically fixed, on the problem, but the real-world applications and the recent developments of various kernel methods have emphasized the need to consider a combination of multiple kernels. A combined kernel can perform a more fine transformation of the initial data space into a larger (with more dimensions) linear separable space.

Following the Lanckriet approach [3], the combined kernel can be obtained as a linear combination of basic kernels:

$$K^* = \sum_{i=1}^k w_i \times K_i \quad (1)$$

where  $k$  represents the cardinal of the considered kernel set,  $K_i$  the  $i^{th}$  kernel and  $w_i$  the weight of the kernel  $K_i$  in the linear combination,  $i \in \{1, \dots, k\}$ . Because the obtained combination must be a SVM kernel function, it has to satisfy the Mercer conditions regarding the positivity and the symmetry of Gram matrix. In [3] is proved that the combination from (1) is a SVM-adapted kernel only if the sum of weights is equal to the number of kernels embedded in that combination and if each weight is less or equal to the number of kernels.

There are two possibilities for choosing these weights: a general approach (see (2)) and a particular approach with non-negative weights (see (3)):

$$w_i \in [-k, k], \text{ with } \sum_{i=1}^k w_i = k \quad (2)$$

$$w_i \in [0, k], \text{ with } \sum_{i=1}^k w_i = k \quad (3)$$

In our model we used a GA [5] for evolving the kernel weights involved into a linear combination. Each GA individual is a fixed-length string of genes. The length of a chromosome is equal to the number of standard kernels that are involved into the combination. Each gene is a real number from  $[0, 1]$  range. Because each gene must be associated with a kernel weight, several transformations must be performed:

1. Transform each gene  $g_i$  into a gene-weight  $gw_i$  using the formula:

$$gw_i = \frac{g_i}{\sum_{j=1}^k g_j}, i = \overline{1, k} \quad (4)$$

2. Scale each gene-weight into the corresponding weight domain. Here, we used two approaches that correspond to those presented by (2) and (3):

- for obtaining the correct weights of a general linear combination with coefficients (negative or positive) whose sum is equal to  $k$ , we must scale each gene-weight using the formula:

$$w_i = -k + gw_i \times 2 \times k, i = \overline{1, k} \quad (5)$$

- for obtaining the correct weights of a particular linear combination with non-negative coefficients whose sum is equal to  $k$ , we must scale each gene-weight using the formula:

$$w_i = gw_i \times k, i = \overline{1, k} \quad (6)$$

### 3.2 Model

The proposed approach is a hybrid technique structured on two levels: a macro level and a micro level. The macro level is a GA that evolves the kernel weights for a combined kernel. The micro level is a SVM algorithm used for computing the quality of a GA individual on the validation dataset.

When we compute the quality of a GA chromosome we actually have to compute the quality of the combined kernel whose weights are encoded in that individual. For assessing the performance of a combined kernel we have to embed that kernel within a SVM algorithm and we have to run the obtained algorithm for a particular problem (classification problem in our case). The accuracy rate computed by the SVM algorithm (on the validation set) represents the quality of a GA individual.

### 3.3 Algorithms

The algorithms used for evolving the kernel coefficients are described in this section. As we said before, we are dealing with a hybrid technique which has a macro level and a micro level.

*Macro-level Algorithm.* The macro level algorithm is a standard GA [5] used for evolving the kernel's coefficients. We use steady-state evolutionary model [8] as underlying mechanism for our GA implementation. The GA starts by creating a random population of individuals. The following steps are repeated until a given number of generations is reached: two parents are selected by using a standard selection procedure. The parents are recombined in order to obtain two offspring by using an uniform arithmetical crossover [9]. The offspring are considered for a Gaussian mutation [10], [11]. The best offspring  $O$  replaces the worst individual  $W$  in the current population if  $O$  is better than  $W$ .

*Micro-level Algorithm.* The micro level algorithm is a SVM algorithm [12], [13], [14] used for computing the fitness of each GA individual from the macro level. The SVM algorithm [1] solves a binary classification problem. As we know, the

<sup>1</sup> It is taken from libsvm [12].

method can be easily extended to multi-classes classification problems, but we performed the numerical experiments for binary classification problems because our approach could be an alternative to Lanckriet approach [3] and we wanted to performed a fairly comparison of the obtained results.

Original implementation of the SVM algorithm from [12] uses one out of the standard kernels (linear, polynomial and Radial Basis Function (RBF) - see Table I). In more cases, the choice of a kernel is empirically performed and it is not problem-adapted. The algorithm proposed in this paper uses a combined kernel (whose coefficients are encoded in the GA chromosome) and a modified version of SVM implementation proposed in [12]. For computing the quality of each GA individual, we run the SVM embedding the combined kernel. The accuracy rate obtain by the SVM algorithm on the validation dataset will represent the quality of that combined kernel. Figure 1 presents the main structure of our model.

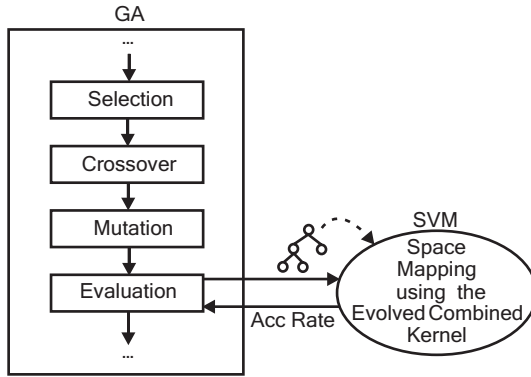


Fig. 1. The sketch of the hybrid approach

## 4 Experiments

### 4.1 Test Problems

The datasets were obtained from UCI repository [15]. Each dataset contains instances labelled with one of two class labels (we must solve a binary classification problem). The *breast* dataset contains 683 instances, the *heart* 270 instances, the *ionosphere* 351 instances and the *sonar* dataset contains 208 instances. Each data set was randomly divided into two sub-sets: learning sub-set (80%) and testing sub-set (20%). The learning sub-set was randomly partitioned into training (2/3) and validation (1/3) parts. In Fig. 2 is depicted this partitioning of a dataset.

The SVM algorithm uses the training subset for learning the model of the SVM and the validation subset for computing the accuracy rate. The best GA chromosome - the best evolved combined kernel ( $ECK^*$ ) - found during learning stage is tested by running a SVM algorithm on the test subset.

Train 2/3 of 80%	Validation 1/3 of 80%	Test 20%
---------------------	--------------------------	-------------

**Fig. 2.** Partitioning of a dataset

## 4.2 Numerical Experiments

**Experiment 1.** In this experiment, a single-objective GA is used for evolving the kernel coefficients. We used a real encoding for GA chromosomes representation that corresponds to the details presented into Sect. 3.1. Also, the transformation presented in that section (see (4), (5) and (6)) are performed in order to obtain valid kernel weights within the corresponding range.

We used a GA population with 20 individuals that are evolved during 50 generations. For obtaining a new generation, we performed a binary tournament selection, a convex crossover and a Gaussian mutation. The values for crossover and mutation probabilities ( $p_c = 0.8$  and  $p_m = 0.1$ ) were chosen for ensuring a good diversity of the population. The crossover and mutation type is specific for real encoding and the values used for the population size and for the number of generation were empirically chosen based on the best results obtained during several tests performed with different values for these parameters.

The standard kernels embedded into the combined kernel  $K^*$  are the linear kernel, the polynomial kernel and the Radial Basis Function (RBF) kernel. These kernels (and their parameters) were chosen for comparison purposes with the results obtained by Lanckriet. Therefore, each GA chromosome will contain three genes, one for each kernel weight. Table 1 contains the expressions of these kernels and the values of their parameters used in our experiments.

**Table 1.** Kernels parameters

Kernel name	Kernel expression	Kernel parameters
Linear	$K_1(u, v) = u^T \times v$	-
Polynomial	$K_2(u, v) = (\gamma \times u^T \times v + coef)^d$	$\gamma = 0.1, coef = 1, d = 3$
RBF	$K_3(u, v) = \exp -\gamma \times  u - v ^2$	$\gamma = 0.1$
ECK	$K^*(u, v) = \sum_{i=1}^k w_i \times K_i(u, v)$	$k = 3, w_i \in [a, b]$

Two sets of experiments were performed for evolving the weights  $w_i$ , ( $i \in \{1, 2, 3\}$ ). In each experiment, we choose a different range for these weights:

- $w_i \in [0, k]$  and  $\sum_i^k w_i = k$  (obtained according to the transformations presented in (4) and (6)) - we denote the obtained evolved combined kernel with  $ECK_{[0,k]}$ ;



- $w_i \in [-k, k]$  and  $\sum_i^k w_i = k$  (obtained according to the transformation presented in (4) and (5)) - we denote the obtained evolved combined kernel with  $ECK_{[-k,k]}$ .

The accuracy rates obtained on each test dataset and for each kernel are presented in Table 2. The results from columns that correspond to different ECKs represent the accuracy rate obtained by running a SVM algorithm (on the test dataset) which used the ECK encoded into the best GA individual (found during learning stage). In this experiment, each SVM algorithm (with a standard or with a combined kernel) is trained and tested on the same dataset.

**Table 2.** The accuracy rate ( $Acc$ ) computed on the test dataset (TDS) by a SVM algorithm that uses different kernels (K):  $K_1, K_2, K_3, ECK_{[0,k]}^*$  and  $ECK_{[-k,k]}^*$ , respectively.  $ECK_{[0,k]}^*$  and  $ECK_{[-k,k]}^*$  mean the best  $ECK_{[0,k]}$  and  $ECK_{[-k,k]}$  obtained during the learning stage. The weights of the ECKs are also given.

Dataset	$K_1$	$K_2$	$K_3$	$ECK_{[0,k]}^*$	$ECK_{[-k,k]}^*$	
breast	98.5401	98.5401	97.8102	98.5401	100	$Acc(\%)$
	-	-	-	1.30/0.25/1.45	2.02/-0.11/1.08	weights
heart	83.3333	81.4815	79.6296	85.1852	87.037	$Acc(\%)$
	-	-	-	0.27/2.03/0.70	-1.27/2.23/2.04	weights
ionosphere	97.1831	95.7746	84.507	98.5915	98.5915	$Acc(\%)$
	-	-	-	0.03/0.08/2.89	0.90/1.78/0.32	weights
sonar	57.1429	61.9048	0	61.9048	76.1905	$Acc(\%)$
	-	-	-	0.35/0.53/2.12	1.51/-0.02/1.51	weights

The results from Table 2 show that ECK performs better than a simple one.  $ECK_{[0,k]}^*$  outperforms the standard kernels for two datasets and equalize the performance of a particular kernel for the other two problems (*breast* and *sonar*). Unlike this kernel, the  $ECK_{[-k,k]}^*$  outperforms the standard kernels for all datasets (in *breast* case it classifies correctly all the test dataset instances).

Moreover, the evolved combined kernel whose weights are from a large range ( $ECK_{[-k,k]}^*$ ) performs better than the evolved combined kernel whose weights are in  $[0, k]$  range ( $ECK_{[0,k]}^*$ ) because it allows a better adaptability of the combined kernel to the classification problem (the  $[-k, k]$  search space is larger than  $[0, k]$  interval =- from the computer point of view).

**Experiment 2.** We also compared our results against those obtained by Lanckriet [3] with a combined kernel learnt with convex methods (CCK). We computed the performance improvement for each dataset and for each kernel type as a percent difference  $\Delta$  between the accuracy rate computed by the SVM algorithm with a combined kernel ( $Acc_{CCK}$ ) - evolved or not - and the accuracy rate computed by an algorithm with a standard kernel ( $Acc_{SK_i}$ ):

$$\Delta_i = \frac{Acc_{CK} - Acc_{SK_i}}{Acc_{SK_i}}, i = \overline{1, k} \quad (7)$$

The obtained differences (from Table 3) show that the  $ECK_{[-k, k]}^*$  improves the SVM performance in all cases, unlike the  $CK_{[-k, k]}$  which decreases the SVM performance in two cases (*breast* and *sonar*). Moreover, the ECK performance improvements are better than those obtained with the CCK in 6 cases (out of 12).

**Table 3.** The  $\Delta$  values. The table presents comparatively the performance improvement of the combined kernels with general weights ( $ECK_{[-k, k]}^*$  and  $CK_{[-k, k]}$ ).

Dataset	$ECK_{[-k, k]}^*$			$CK_{[-k, k]}$		
	$K_1$	$K_2$	$K_3$	$K_1$	$K_2$	$K_3$
breast	1%	1%	2%	9%	-1%	7%
heart	4%	7%	9%	1%	7%	43%
ionosphere	1%	3%	17%	14%	0%	3%
sonar	33%	23%	0%	15%	8%	-1%

**Experiment 3.** For a complete analysis of the ECKs performance, we tested the generalization ability of the ECKs. For this experiment we use the one of the previous ECKs obtained for a particular dataset. This combined kernel is embedded into a SVM algorithm which is run against other 3 datasets. The results obtained by running the SVM with the ECK based on *sonar* dataset – for both weight types (second column and third column) – and with the standard kernels (next three columns) are presented in Table 4.

**Table 4.** Generalization ability of the evolved kernel. *sonar* dataset [15] was used as training set and the other datasets were used as test dataset.

Dataset	$ECK_{[0, k]}^*$	$ECK_{[-k, k]}^*$	$K_1$	$K_2$	$K_3$
breast	97.8102	99.2701	98.5401	98.5401	97.8102
heart	79.6296	85.1852	83.3333	81.4815	79.6296
ionosphere	97.1831	92.9577	97.1831	95.7746	84.5070
sonar	<b>61.9048</b>	<b>76.1905</b>	57.1429	61.9048	0.0000

Table 4 show that the  $ECK_{[0, k]}^*$  performs similarly with the standard kernels in 2 cases, but the  $ECK_{[-k, k]}^*$  increases the SVM performance for all problems and for all kernel pairs (evolved, standard). Therefore, these results can be considered an empirically proof of the good ECKs generalization ability.

## 5 Conclusions

A hybrid technique for evolving kernel weights has been proposed in this paper. The model has been used for evolving the weights of a combined kernel embedded into a SVM algorithm that solves a binary classification problem.

We have performed several numerical experiments for comparing our ECKs to others kernels (simple and combined – whose weights are determined by convex methods). Numerical experiments have shown that the ECKs perform similarly and sometimes even better than the standard kernels (linear, polynomial and RBF kernels). Moreover, the ECKs performance also outperform the CCKs (obtain in [3]) for some of the test problems. Our evolved kernels are also more robust in comparison with that of Lanckriet (which works worst than standard kernels in some cases).

However, taking into account the No Free Lunch theorems for Search [16] and Optimization [17] we cannot make any assumption about the generalization ability of the evolved kernel weights. Further numerical experiments are required in order to assess the power of the evolved kernels.

Further work will be focused on:

- evolving better-combined kernels
- using multiple data sets or a multi-objective evolutionary approach for the training stage
- using Genetic Programming [18] technique for obtained more complex combined kernels.

## References

1. Joachims, T.: Making large-scale SVM learning practical. In: *Advances in Kernel Methods — Support Vector Learning*, MIT Press (1999) 169–184
2. Platt, J.: Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, MSR (1998)
3. Lanckriet, G.R.G., et al.: Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research* **5** (2004) 27–72
4. Fogel, D.B.: *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press (1995)
5. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley (1989)
6. Sonnenburg, S., Rtsch, G., Schfer, C., Schlkopf, B.: Large scale multiple kernel learning. *Journal of Machine Learning Research* **7** (2006) 1531–1565
7. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the SMO algorithm. In: *Machine Learning, Proceedings of ICML 2004*, ACM (2004)
8. Syswerda, G.: A study of reproduction in generational and steady state genetic algorithms. In: *Proc. of FOGA*, Morgan Kaufmann Publishers (1991) 94–101
9. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer (1992)
10. Yao, X., Liu, Y., Lin, G.: Evolutionary Programming made faster. *IEEE-EC* **3**(2) (1999) 82

11. Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence through Simulated Evolution. John Wiley & Sons (1966)
12. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
13. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (2000)
14. Vapnik, V.: Statistical Learning Theory. Wiley (1998)
15. Newman, D., Hettich, S., Blake, C., Merz, C.: Uci repository of machine learning databases (1998)
16. Wolpert, D.H., Macready, W.G.: No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute (1995)
17. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation **1**(1) (1997) 67–82
18. Koza, J.R.: Genetic programming II: automatic discovery of reusable programs. MIT Press (1994)

# Boosting RVM Classifiers for Large Data Sets

Catarina Silva<sup>1,2</sup>, Bernardete Ribeiro<sup>2</sup>, and Andrew H. Sung<sup>3</sup>

<sup>1</sup> School of Technology and Management, Polytechnic Institute of Leiria, Portugal

<sup>2</sup> Dep. Informatics Eng., Center Informatics and Systems, Univ. of Coimbra, Portugal  
{catarina, bribeiro}@dei.uc.pt

<sup>3</sup> Dep. Comp. Science, Inst. Complex Additive Sys. Analysis, New Mexico Tech, USA  
sung@cs.nmt.edu

**Abstract.** Relevance Vector Machines (RVM) extend Support Vector Machines (SVM) to have probabilistic interpretations, to build sparse training models with fewer basis functions (i.e., relevance vectors or prototypes), and to realize Bayesian learning by placing priors over parameters (i.e., introducing hyperparameters). However, RVM algorithms do not scale up to large data sets. To overcome this problem, in this paper we propose a RVM boosting algorithm and demonstrate its potential with a text mining application. The idea is to build weaker classifiers, and then improve overall accuracy by using a boosting technique for document classification. The algorithm proposed is able to incorporate all the training data available; when combined with sampling techniques for choosing the working set, the boosted learning machine is able to attain high accuracy. Experiments on REUTERS benchmark show that the results achieve competitive accuracy against state-of-the-art SVM; meanwhile, the sparser solution found allows real-time implementations.

## 1 Introduction

Relevance Vector Machines (RVM) are a powerful form of Bayesian inference based on Gaussian Processes that is becoming widespread in the machine learning community. The RVM introduced by Tipping [1] can also be viewed as a Bayesian framework of the kernel machine setting, producing an identical functional form to the well-known SVM. Tipping compared SVM and RVM and showed that while maintaining comparable performance, RVM requires dramatically less kernel functions than SVM, thus gaining a competitive advantage, especially in real-time applications. However, for a training set of  $N$  examples, RVM training time is  $O(N^3)$  and memory scaling  $O(N^2)$ ; this has become an obstacle to the use of RVM in problems with large data sets. In the literature several attempts have been made to deal with this problem. Williams et al. [2] applied the Nyström method to calculate a reduced rank approximation of the  $n \times n$  kernel matrix. Csató et al. [3] developed an on-line algorithm to maintain a sparse representation of the model. Smola et al. [4] proposed a forward selection scheme to approximate the log posterior probability. Candela [5] suggested a promising alternative criterion by maximizing the approximate model evidence.

Tipping [6] also suggested an incremental learning strategy that starts with only a single basis function and adds basis functions along the iterations.

In this paper we propose a boosting approach to deal with problems involving large data sets and empirically show its effectiveness in a text classification problem. Boosting techniques generate many relatively weak classification rules and combine them into a single highly accurate classification rule. In particular AdaBoost [7] solved many of the practical difficulties of the earlier boosting algorithms.

We present a new RVM boosting version which is able to take into account all the information available for RVM training, using boosting to generate an overall model. High-dimensionality problems prone to RVM are tackled by randomly choosing working sets that are then boosted repeatedly.

Next section briefly reviews the RVM to lay the foundations of the proposed approach. Section 3 introduces the boosting concept, exploring in some detail the AdaBoost algorithm. Section 4 presents the proposed RVM boosting algorithm. In Sect. 5 the experimental setting is described, namely the test collection, effectiveness metric and feature selection as well as the performance results on the text mining application. In Sect. 6 conclusions and future work are addressed.

## 2 Relevance Vector Machines

The RVM was proposed by Tipping [1], as a Bayesian treatment of the sparse learning problem. The RVM preserves the generalization and sparsity of the SVM, yet it also yields a probabilistic output, as well as circumvents other limitations of SVM, such as the need of Mercer kernels and the definition of the error/margin trade-off parameter  $C$ . The output of an RVM model is very similar to the Vapnik proposed SVM model [8], and can be represented as:

$$g(\mathbf{x}) = \sum_{i=1}^N w_i k(\mathbf{x}, \mathbf{z}_i) + w_0, \quad (1)$$

where  $\mathbf{x}$  is an input vector and  $g : \mathbb{R}^M \rightarrow \mathbb{R}$  is the scalar-valued output function, modeled as a weighted sum of kernel evaluations between the test vector and the training examples. The kernel function,  $k : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}$  can be considered either as a similarity function between vectors, or as a basis function centered at  $\mathbf{z}_i$ . Training determines the weights,  $\mathbf{w} = [w_0, \dots, w_N]$  while the sparsity property will rise if some of the  $w_i$  are set to zero.

Applying Bayesian inference, a prior distribution is specified over  $\mathbf{w}$ , defined by Tipping as a zero-mean Gaussian distribution  $p(\mathbf{w}) = N(\mathbf{w}; 0, A)$ , where  $N(\mathbf{d}; e, F)$  denotes the multivariate Gaussian distribution of a random vector  $\mathbf{d}$  with mean  $e$  and covariance matrix  $F$ . The covariance  $A = \text{diag}(\alpha_0, \dots, \alpha_N)$  has an individual and independent hyperparameter for every weight. It is assumed that the training targets are sampled with additive noise:  $t_i = g(\mathbf{z}_i) + \epsilon_i$ , where  $\epsilon_i \sim N(\epsilon; 0, \sigma^2)$ . Thus the likelihood becomes  $p(t_i | \{\mathbf{z}_i\}) = N(t_i; g(\mathbf{z}_i), \sigma^2)$ , where  $\sigma^2$  is another hyperparameter, but it is global across the training set.

Without considering hyperpriors, the posterior is also Gaussian:

$$p(\mathbf{w}|\{\mathbf{z}_i, t_i\}, \alpha, \sigma^2) = N(\mathbf{w}; \hat{\mathbf{w}}, \Sigma), \quad (2)$$

with  $\Sigma^{-1} = \sigma^{-2}\Phi^T\Phi + A$  and  $\hat{\mathbf{w}} = \sigma^{-2}\Sigma\Phi^T\mathbf{t}$ .  $\Phi$  is called the *design matrix* and contains the intratraining set kernels values:  $\Phi_{ij} = k(\mathbf{z}_i, \mathbf{z}_j)$ . As shown in (2),  $\hat{\mathbf{w}}$  and  $\Sigma$  are determined by the hyperparameters  $\alpha, \sigma^2$ , which characterize different models and training involves finding these values via Bayesian model selection. To do this, the *evidence*,  $p(\{\mathbf{z}_i, t_i\}|\alpha, \sigma^2)$  is maximized using a variant of gradient ascent. The algorithm begins optimizing over the entire training set, but examples are pruned whenever the associated  $\alpha_i$  falls below a threshold, leading to the final sparse solution. Those examples remaining with  $w_i \neq 0$  are termed *relevance vectors*. Figure 1 shows a two dimensional RVM classification example with four *relevance vectors*. Each iteration of the training algorithm evaluates the posterior (2), which involves matrix inversion: a  $O(N^3)$  cost operation.

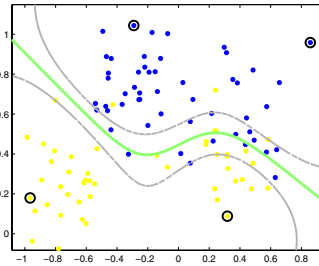


Fig. 1. RVM classification example

### 3 Boosting

This section introduces the boosting concept and details the AdaBoost algorithm proposed by Freund and Schapire [7], which is the base of the proposed RVM Boosting method, presented in the following section.

The main idea of boosting is to generate many relatively weak classification rules and to combine them into a single highly accurate classification rule. Boosting algorithms have some interesting features, as it will be clear throughout the paper. It has been shown to perform well experimentally on several learning tasks, such as text classification [9].

The weak classification rules are more formally denominated *weak hypothesis*. Boosting assumes there is an algorithm for generating them, called the *weak learning algorithm*. The boosting algorithm calls the weak learner as often as needed to generate many hypothesis that are combined resulting into a *final* or *combined hypothesis*.

One distinctive feature of the boosting algorithm is that, during its progress, it assigns different importance weights to different training examples. The algorithm proceeds by incrementally assigning greater significance to harder to

classify examples, while easier training examples get lower weights. This weight strategy is the basis of the weak learners evaluation. The final combined hypothesis classifies a new test example by computing the prediction of each of the weak hypothesis and taking a vote on these predictions. The algorithm starts with  $N$  input-target pairs  $\langle (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N) \rangle$ , where  $\mathbf{x}_i$  is a training example and  $y_i \in \{-1, +1\}$  is the associated label, usually defined by a human expert. Initially the importance weights of the examples are uniformly distributed ( $X_1(i) = \frac{1}{N}$ ). Then, AdaBoost algorithm repeatedly retrieves *weak hypothesis* that are evaluated and used to determine the final hypothesis. On iteration  $s$ , using the set of importance weights determined on iteration  $s - 1$ , the hypothesis error,  $\epsilon_s$ , is computed and  $\alpha$ , representing the weight or importance of that *weak classifier*, is determined. The proposed expression for  $\alpha$  assigns larger weights to *good classifiers*, i.e., classifiers with low error, and lower weights (even with negative values) for *bad classifiers*. In the unlikely event of the hypothesis error,  $\epsilon_s$ , being zero, we are in the presence of a very poor classifier that misclassifies all training examples and that should be discarded. Before a new iteration begins, the importance weight of each example is updated as a distribution over the training examples.

## 4 Boosting RVM

Nevertheless RVM may not be considered *weak learners*, we empirically show that the boosting concept can be applied as a way of circumventing RVM scaling problems. The main idea in our RVM boosting is to use all the training examples, by sampling them into small working sets, making each classifier much weaker than it would be if trained with all available training examples. If enough models are generated, all distinctive aspects of the class are captured and represented in the final classifier.

Dividing the huge data set into smaller tractable chunks, the computational load usually associated with training RVMs is mitigated. Moreover, due to the independence of the *not so weak* RVM classifiers, it is possible to distribute the computational burden in a cluster or other distributed environment.

Our major innovations were performed to adapt the AdaBoost algorithm to RVM boosting. First, instead of using the training set for training and for boosting, a separate boosting set was defined. Second, as the RVM classifiers are in fact *not so weak classifiers*, as the AdaBoost assumes, the same set of classifiers was presented repeatedly to the boosting algorithm, i.e., the number of iterations is not equal to the number of classifiers, but it is proportional. Algorithm [1](#) shows the changes carried out to obtain the RVM boosting algorithm. Our idea to consider different training and boosting sets is justifiable in large scale problems, since there are enough examples to spare. Also when the training sets are large and sparse, convergence problems may occur by boosting the classifier with the same set. These convergence problems can lead to the algorithm's inability to define which are the harder examples, ie, with larger weight in the classifier evaluation, given that using the same learning and boosting sets may result in insufficient diversity.



---

**Algorithm 1.** RVM boosting algorithm

---

**Input:** $N$  training labeled examples:

$$\langle (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N) \rangle, \text{ where } y_i \in \{-1, +1\}$$

 $N_{boost}$  boosting labeled examples:

$$\langle (\mathbf{x}_{N+1}, y_{N+1}), \dots, (\mathbf{x}_{N+N_{boost}}, y_{N+N_{boost}}) \rangle, \text{ where } y_i \in \{-1, +1\}$$

integer  $NC$  specifying the number of classifiersinteger  $T$  specifying the number of iterations**Initialize**  $X_1(i) = \frac{1}{N_{boost}}$ **for**  $s = 1, 2, \dots, T$  **do** $c = s \bmod NC$ **if**  $c = 0$  **then** $c = NC$ **end if**Call weak learner and get weak hypothesis  $h_c$ Calculate the error of  $h_s$ :  $\epsilon_s = \sum_{i: h_c(x_i) \neq y_i} X_s(i)$ Set  $\alpha_s = \frac{1}{2} \ln \left( \frac{1 - \epsilon_s}{\epsilon_s} \right)$ 

Update distribution:

$$\begin{aligned} X_{s+1}(i) &= \frac{X_s(i) e^{-\alpha_s y_i h_s(x_i)}}{Z_s} \\ &= \frac{X_s(i)}{Z_s} \times e^{-\alpha_s}, \text{ if } h_c(x_i) = y_i \\ &= \frac{X_s(i)}{Z_s} \times e^{\alpha_s}, \text{ if } h_c(x_i) \neq y_i \end{aligned}$$

where  $Z_s$  is a normalization factor.**end for****Output:** the final hypothesis:

$$h_{fin}(x) = \text{sign} \left( \sum_{s=1}^T \alpha_s h_c(x) \right).$$

---

## 5 Case Study: Text Classification

To evaluate the proposed RVM boosting algorithm, a large scale problem was chosen: text mining/classification, i.e., automatically assign semantic categories to natural language text. In the last two decades the production of textual documents in digital form has increased exponentially [10], consequently there is an ever-increasing need for automated solutions for organizing the huge amount of digital texts produced. Documents are represented by vectors of numeric values, with one value for each word that appears in any training document, making it a large scale problem. High dimensionality increases both processing time and the risk of overfitting. In this case, the learning algorithm will induce a classifier that reflects accidental properties of the particular training examples rather than the systematic relationships between the words and the categories [11]. To deal with this dimensionality problem, feature selection and dimension reduction methods

are applied, such as, stopword removal, stemming and removing less frequent words. Yang [15] presents a noteworthy scalability analysis of classifiers in text mining, including KNN and SVM.

### 5.1 Data Set

For the experiments, REUTERS-21578 dataset [1] was used. It is a financial corpus with news articles averaging 200 words each. REUTERS-21578 corpus has about 21578 classified stories into 118 possible categories. We use only 10 categories (earn, acq, money-fx, grain, crude, trade, interest, ship, wheat and corn), detailed in Table 1 with the corresponding number of positive training and testing examples, since they cover 75% of the items and constitute an accepted benchmark. The ModApte split was used, using 75% of the articles (9603 items) for training and 25% (3299 items) for testing.

**Table 1.** Number of positive training and testing documents for REUTERS-21578 most frequent categories

Category	Training	Testing	Category	Training	Testing
earn	2715	1044	trade	346	113
acq	1547	680	interest	313	121
money-fx	496	161	ship	186	89
grain	395	138	wheat	194	66
crude	358	176	corn	164	52

### 5.2 Performance Measures

In order to evaluate a binary decision task we first define a contingency matrix representing the possible outcomes of the classification, as shown in Table 2. Several measures have been defined based on this contingency table, such as, Error Rate ( $\frac{b+c}{a+b+c+d}$ ), Recall ( $\frac{a}{a+b}$ ) and Precision ( $\frac{a}{a+c}$ ). Measures that combine recall and precision have been defined: break-even point (BEP) and  $F_\beta$  measure. BEP was proposed by Lewis [12] and is defined as the point at which recall equals precision. van Rijsbergen’s  $F_\beta$  measure [13] combines recall and precision in a single score [3]:

$$F_\beta = \frac{(\beta^2 + 1)P \times R}{\beta^2 P + R} = \frac{(\beta^2 + 1)a}{(\beta^2 + 1)a + b + \beta^2 c} \tag{3}$$

$F_0$  is the same as Precision,  $F_\infty$  is the same as Recall. Intermediate values between 0 and  $\infty$  correspond to are different weights assigned to recall and precision. The most common values assigned to  $\beta$  are 0.5 (recall is half as important as precision), 1.0 (recall and precision are equally important) and 2.0 (recall is twice as important as precision). If  $a$ ,  $b$  and  $c$  are all 0,  $F_\beta$  is defined as 1 (this

<sup>1</sup> <http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>

**Table 2.** Contingency table for binary classification

	Class Positive	Class Negative
Assigned Positive	a (True Positives)	b (False Positives)
Assigned Negative	c (False Negatives)	d (True Negatives)

occurs when a classifier assigns no documents to the category and there are no related documents in the collection).

None of the measures is perfect or appropriate for every problem. For example, recall, if used alone might show deceiving results, e.g., a system that classifies all testing examples as belonging to a given category will show perfect recall, since measure  $c$  in Table 2 will be zero, making recall ( $\frac{a}{a+c}$ ) reach its maximum. Accuracy (1-Error Rate), on the other hand works well if the number of positive and negative examples are balanced, but in extreme conditions it might be deceiving too. If the number of negative examples is overwhelming compared to the positive examples, as in text classification, then a system that assigns no documents to the category will obtain an accuracy value close to 1.

van Rijsbergens F measure is the best suited measure, but still has the drawback that it might be difficult for the user to define the relative importance of recall and precision [14]. In general the F1 performance is reported as an average value. There are two ways of computing this average: macro-average, and micro-average. With macro-average the F1 value is computed for each category and these are averaged to get the final macro-averaged F1. With micro-average we first obtain the global values for the true positive, true negative, false positive, and false negative decisions and then compute the micro-averaged F1 value using the micro-recall and micro-precision (computed with these global values). The results reported in this paper are macro-averaged F1. This allows us to compare our results with those of other researchers working with Reuters dataset.

### 5.3 Baseline Results

To serve as a baseline of comparison, Table 3 presents the summary of the results achieved with linear RVM for REUTERS-21578. Only the words that appear in more than 500 documents, i.e., with a Document Frequency (DF) over 500 ( $DF > 500$ ) resulting in 176 words, were used in each setting. This initial feature selection was carried out to reduce the dimension of the problem. The document frequency threshold was increased stepwise while there was no severe classification performance compromise. Training was carried out with 1000 and 2000 documents. Settings using more features and more training examples were not attainable in reasonable computational time.

The results are represented in terms of F1 values where classification performance is concerned and in terms of number of Relevance Vectors (RV) and CPU training time (in seconds) to determine the computational complexity of the solution.

**Table 3.** F1, Relevance Vectors and CPU training time (in seconds) for Baseline RVM

	1000 documents			2000 documents		
	F1	RV	CPU	F1	RV	CPU
earn	94.65%	23	111	95.95%	36	828
acq	87.72%	25	93	89.80%	41	616
money-fx	46.43%	16	81	57.25%	34	626
grain	77.58%	19	126	80.61%	25	849
crude	68.17%	19	78	72.90%	27	606
trade	48.28%	20	104	43.96%	25	558
interest	59.65%	13	78	54.44%	22	584
ship	37.84%	15	98	62.28%	22	583
wheat	81.43%	12	83	79.14%	14	503
corn	57.45%	14	112	62.14%	23	542
average	65.92%	17.6	96.4	69.85%	26.9	629.5

#### 5.4 RVM Text Boosting

This section presents the deployment to text classification of the RVM boosting method detailed in Sect. 4. For REUTERS-21578, 1000 boosting examples were considered. To make comparable predictions with the baseline results, the *weak classifiers* will be generated using random samples of 1000 and 2000 documents, taken from the remaining training examples. Thus, two settings tested. One using 20 classifiers trained with 1000 examples and the other using also 20 classifiers, but trained with 2000 training examples. In both settings several values for the number of iterations were tested, but always proportional to the number of classifiers (20 in both settings). To develop the boosting algorithm two sets of 20 RVM models were built. Initially a set of 1000 documents retrieved from the training set was kept for the following task of boosting the sets of classifiers.

To train the first set, 1000 documents were each time randomly chosen from the rest of the training examples; and for the second set, 2000 documents were

**Table 4.** F1 results for boosting 20 classifiers of 1000 documents

Iterations	20	40	60	80	100	120	140	160
earn	95.96%	96.78%	96.83%	96.83%	96.97%	96.88%	96.88%	96.83%
acq	88.63%	90.69%	90.82%	91.24%	91.10%	90.94%	91.19%	91.24%
money-fx	54.62%	57.79%	59.85%	60.67%	61.65%	61.36%	61.42%	60.52%
grain	75.97%	77.10%	79.69%	77.44%	77.15%	77.61%	77.61%	76.87%
crude	68.99%	67.61%	69.96%	70.46%	70.61%	69.78%	68.84%	67.39%
trade	62.44%	63.37%	63.37%	61.39%	62.07%	62.75%	63.46%	61.84%
interest	56.60%	57.86%	59.39%	59.88%	62.79%	61.27%	61.63%	60.82%
ship	67.53%	67.53%	66.67%	67.95%	70.20%	70.20%	70.20%	69.74%
wheat	84.06%	83.21%	82.86%	82.01%	82.01%	81.43%	80.85%	80.58%
corn	63.83%	63.83%	63.83%	63.04%	66.67%	67.44%	66.67%	66.67%
average	71.86%	72.58%	73.33%	73.09%	74.13%	73.97%	73.88%	73.25%

**Table 5.** F1 results for boosting 20 classifiers of 2000 documents

Iterations	20	40	60	80	100	120	140	160
earn	95.17%	96.34%	96.75%	96.70%	96.74%	96.79%	96.84%	96.79%
acq	88.26%	89.86%	90.20%	89.99%	89.79%	89.52%	89.68%	89.50%
money-fx	61.64%	60.28%	60.99%	62.37%	63.04%	63.94%	63.67%	62.12%
grain	78.13%	77.82%	79.68%	81.12%	80.31%	81.57%	82.03%	81.71%
crude	73.33%	74.55%	76.04%	76.53%	76.77%	76.77%	76.77%	77.42%
trade	58.72%	64.84%	63.30%	63.59%	63.01%	63.01%	63.01%	61.88%
interest	53.55%	56.38%	62.31%	62.94%	65.61%	65.26%	67.02%	66.67%
ship	57.14%	57.93%	57.14%	57.93%	58.50%	61.22%	61.22%	60.81%
wheat	84.21%	85.07%	87.22%	87.22%	87.22%	87.22%	86.36%	86.36%
corn	61.54%	61.39%	61.39%	60.78%	61.39%	62.75%	62.75%	62.75%
average	71.17%	72.45%	73.50%	73.92%	74.24%	74.81%	74.94%	74.60%

chosen the same way. Analysing the results presented in tables 4 and 5, there is an improvement when the number of iterations rises to around 100-140, showing that more randomly generated classifiers could improve the overall performance. The improvement achieved with this simple boosting schema is of 8% for the 1000 documents classifiers and 5% for the 2000 documents classifiers.

## 6 Conclusions and Future Work

We have developed an efficient RVM boosting algorithm for large data sets, taking into account all the information available for RVM training, using boosting to generate an overall model. High-dimensionality problems prone to RVM are tackled by randomly choosing working sets that then are boosted repeatedly.

Two major changes were incorporated to adapt the AdaBoost algorithm for RVM boosting. First, instead of using the training set for training and for boosting, a separate boosting set was defined. Second, as the RVM classifiers are in fact *not so weak classifiers*, the same set of classifiers was presented repeatedly to the boosting algorithm, i.e., the number of iterations is not equal to the number of classifiers, but it is proportional. RVM boosting effectiveness was empirically tested on a text classification problem, improving baseline results while maintaining probabilistic outputs and sparse decision functions.

Future work will deal with the refinement of the boosting method, namely the distribution measure, which refers to the weight or significance assigned to each base classifier, could be differently aligned with the training data.

**Acknowledgments.** Portuguese Foundation for Science and Technology through Project POSI/SRI/ 41234/2001 and CISUC - Center of Informatics and Systems of University of Coimbra - are gratefully acknowledged for partial financing support.

## References

1. M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine", *Journal of Machine Learning Research I*, 2001, pp 211-214.
2. M. Seeger, C. Williams, N. Lawrence, "Fast Forward Selection to Speed up Sparse Gaussian Process Regression", *International Workshop on AI and Statistics*, 2003.
3. L. Csató, M. Oppor, "Sparse Online Gaussian Processes", *Neural Computation*, Vol.14, pp.641-668, 2002.
4. A. Smola, P. Bartlett, "Sparse Greedy Gaussian Processes Regression", *Advances in Neural Information Processing 13*, pp. 619-625, 2001.
5. J. Candela, "Learning with Uncertainty - Gaussian Processes and Relevance Vector Machines", PhD thesis, Technical University of Denmark, 2004.
6. M. Tipping, A. Faul, "Fast Marginal Likelihood Maximisation for Sparse Bayesian Models", *International Workshop on Artificial Intelligence and Statistics*, 2003.
7. Y. Freund, R. Schapire, "Experiments with a new boosting algorithm", *International Conference Machine Learning*, pp. 148-156, 1996.
8. V. Vapnik, "The Nature of Statistical Learning Theory", 2nd ed, Springer, 1999.
9. R. Schapire, Y. Singer, "Boostexter: A Boosting-based System for Text Categorization", *Machine Learning*, 39(2/3), pp. 135-168, 2000.
10. F. Sebastiani, "Classification of Text, Automatic", *The Encyclopedia of Language and Linguistics*, In Keith Brown (ed.), Volume 14, 2nd Edition, Elsevier, 2006.
11. S. Eyheramendy, A. Genkin, W. Ju, D. Lewis, D. Madigan, "Sparse Bayesian Classifiers for Text Classification", *Journal of Intelligence Community R&D*, 2003.
12. D. Lewis, "An evaluation of phrasal and clustered representations on a text categorization task", *15th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3750, 1992.
13. C. van Rijsbergen, "Information Retrieval", 2nd ed. Butterworths, London, 1979.
14. M. Ruiz, P. Srinivasan, "Hierarchical Text Categorization Using Neural Networks", *Information Retrieval*, 5, pp. 87118, 2002.
15. Y. Yang, J. Zhang, B. Kisiel, "A Scalability Analysis of Classifiers in Text Categorization", *SIGIR '03*, ACM Press, 2003, pp 96-103.

# Multi-class Support Vector Machines Based on Arranged Decision Graphs and Particle Swarm Optimization for Model Selection

Javier Acevedo, Saturnino Maldonado,  
Philip Siegmann, Sergio Lafuente, and Pedro Gil

University of Alcala, Teoría de la señal,  
Alcala de Henares, Spain  
javier.acevedo@uah.es

<http://www2.uah.es/teose>

**Abstract.** The use of support vector machines for multi-category problems is still an open field to research. Most of the published works use the one-against-rest strategy, but with a one-against-one approach results can be improved. To avoid testing with all the binary classifiers there are some methods like the Decision Directed Acyclic Graph based on a decision tree. In this work we propose an optimization method to improve the performance of the binary classifiers using Particle Swarm Optimization and an automatic method to build the graph that improves the average number of operations needed in the test phase. Results show a good behavior when both ideas are used.

## 1 Introduction

Support vector machines [1] (SVM) have been applied with a satisfactory level of success to many different two-class problems [2]. Based on the Statistical Learning Theory (SLT) SVM try to improve the statistical risk rather than the empirical risk. Due to this reason, SVM give a better performance than other learning machines when classifying unseen patterns.

The extension to the multi-category problems, where there are  $N$  different classes, does not present an easy solution and is still a field to research. In [3] it was exposed a mathematical formulation to extend the binary case to multi-category problems, but it has to deal with all the support vectors at the same time, resulting a complex classifier that does not provide high performance in many problems. Most of the published works make the extension to the multi-class case building  $N$  different classifiers in the so called one-against-rest approach. The usual method is to compare the outputs of the classifiers and to select the one with the highest value. However, in [4] it is remarked that the output of an SVM is not a calibrated value and should be not compared. The way to solve this problem is also proposed in the same paper, adding to the output a estimation of the probability of success.

Another binary based approach is the so called one-against-one approach, where  $N(N - 1)/2$  classifiers are built, each being classifier trained only on

two of the  $N$  classes. Although this approach can give better results than the one-against-rest case, this scheme has not been widely applied due to the fact that the number of classifiers increase exponentially with the number of classes. However, in most of real applications what it really matters is the time needed to compute the test phase, specially in some real time systems. With this approach, in the training phase, the classifiers obtained can be simpler than in the one-against-rest case. In [5] it was proposed the Max Wins algorithm, obtaining very good results, but it implies that in the test phase all the built classifiers should be used, with a high cost from a computational point of view. In [6] the proposal was to build a graph with the binary classifiers, in such a way that in the test phase it is only necessary to work with  $N$  classifiers. This method was called Decision Directed Acyclic Graph based on SVM (DAGSVM). In this work, we propose an automatic method to arrange the graph, resulting in less average time to test the samples.

On the other hand, one of the major problems when using binary classifiers is the choice of the kernels, the parameters associated to these kernels and the value of the regularizing parameter  $C$ . The right choice of these parameters, known as model selection, improves the performance of the binary classifiers in a considerable way. In the proposal of the multi-class method of this work, the success of the binary classifiers is basic to ensure the overall performance. In this paper we have applied Particle Swarm Optimization (PSO) [7] to find the optimal value of the parameters.

## 2 Building the Set of Binary Classifiers

### 2.1 Model Selection

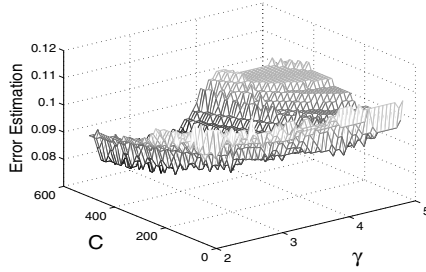
One of the most difficult points in classification is to tune the parameters associated to the learning machine. In our case, when working with SVM, the choice of the kernel is going to have great influence in the success of the classifier. Most of the classification problems are not linearly separable and a kernel method has to be applied. The most popular kernel for non linear cases is the Radial Basis Function One (II).

$$K(\mathbf{x}, \mathbf{y}) = e^{-\gamma \sum_{i=1}^n (x_i - y_i)^2} . \quad (1)$$

When this kernel is used, in addition to the  $C$  parameter, the  $\gamma$  parameter has to be tuned. Instead of using common parameters for all the classifiers, as it has been proposed in the previous mentioned work, it seems more logical to find the best combination of parameters for each binary classifier. In Fig. (II) it can be appreciated how the estimation of the error varies with these two parameters.

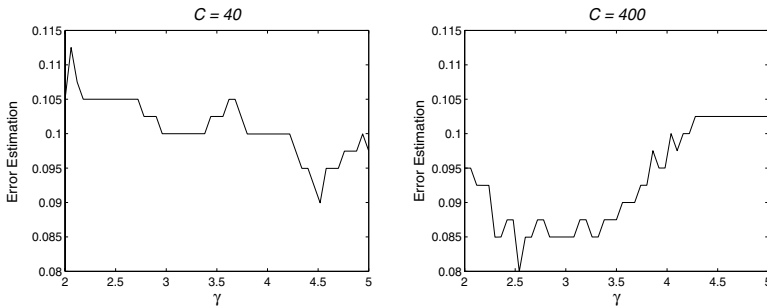
Most of the published works fix a  $C$  parameter and search to find the best  $\gamma$  obtained. Then, having found the  $\gamma$  parameter, the best possible  $C$  parameter is calculated. However, in Fig. (II) it can be appreciated that, fixing the  $C$  parameter to low values in this case, lead us to select a wrong value of the  $\gamma$  parameter. So, it is necessary to take into account both parameters at the same time, searching





**Fig. 1.** An example of the Error with Different values for  $C$  and  $\gamma$

for the combination of them that minimizes the estimation of the error. The direct method is to make the search space discrete and test all the possible combinations, but this procedure is very expensive from a computational point of view, specially when the one-against-one method is selected, due to the high number of classifiers to be used. So, the proposal is to use a statistical search method (SSM) to find an optimal value of the kernel parameters.



**Fig. 2.** An example of the Error Estimation with  $C$  fixed and  $\gamma$  variable

There are several estimators of the generalization error that can be used as evaluation function. The leave-one-out error is known to be an unbiased estimator, but it requires to train many SVM. The most extended estimator, as it has been used in this work, is the k-fold crossvalidation, that gives good results in a reasonable time.

## 2.2 PSO for Tuning SVM Parameters

Once that the importance of the parameter has been exposed, the question is how to select the appropriated values. It has to be noted that the functions described to estimate the error are not derivable and as it is shown in Fig. (1), there are multiple local minima. Moreover, there is not a priori information of the

error function until we train the dataset and the error is estimated. With these starting points, a method based on SSM for continuous function minimization seems to be appropriated to solve our problem.

PSO is a recent method for function minimization and it is inspired by the emergent motion of a flock of birds searching for food. Like in other SSM the search for the optimum is an iterative process that is based on random decisions taken by  $m$  particles searching the space at the same time. Each particle  $i$  has an initial position  $x_i$  that is a vector with a possible solution of the problem. In our case, the components of the vector are the  $C$  and  $\gamma$  parameters if the kernel is RBF. Each position is evaluated with the objective function and the particles update their position according to (2) where  $v_i(t+1)$  is the new velocity of particle  $i$ ,  $\phi(t)$  is the inertia function,  $pbest$  is the best position achieved by the particle  $i$ ,  $gbest$  is the best position achieved by any of the particles,  $c_{1,2}$  are coefficients related to the strength of attraction to the  $pbest$  and  $gbest$  position respectively and  $r_{1,2} \in [0, 1]$  are random numbers.

$$\begin{aligned} v_i(t+1) &= \phi(t) v_i(t) + c_1 r_1 (pbest - x_i) + c_2 r_2 (gbest - x_i) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \quad (2)$$

The search of each particle is done using its past information and the neighborhood one. This fact makes that the particles fly to a minima position but they can scape if it is a local minima.

As it has been mentioned, the evaluation function measures the crossvalidation error. However, it has been observed that in some problems there are several combinations allowing the error to be minimized, usually in a plain region of the crossvalidation error. In such cases, the best choice is to select the solution that also minimizes the number of operations required as it is described in (3).

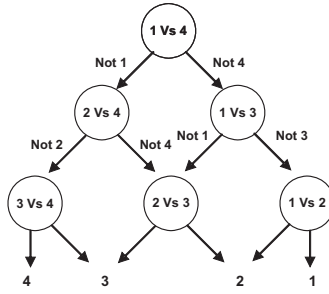
$$f(C, \gamma) = \widehat{Error}(C, \gamma) + \frac{\bar{N}_s}{l^2} \quad (3)$$

Where  $\bar{N}_s$  is the average number of support vector obtained as a result of the crossvalidation partition and  $l$  is the total number of samples available for the training phase.

### 3 Graph Order

Once the classifiers have been trained, each of them with its optimal parameters, we can go back to the DAGSVM algorithm. In Fig.(3) it is shown the proposed order for a 4-classes problem in the DAGSVM method. After testing many datasets it can be said that the order of the graph is not relevant in the total accuracy of the classifier, but it plays a crucial role in the average operations needed in the test phase.

Let us assume that the classifier separating class 1 and class 4 is the one with a high number of operations needed to be evaluated in the test phase. The proposed graph of Fig.(3) makes that all the test samples have to be evaluated through this classifier, but in many cases this classifier is not relevant for the problem.



**Fig. 3.** Graph proposed for 4 classes in the DAGSVM method

If the classifiers have been trained as proposed in the described method to optimize the performance, it is clear that some of the classifiers are best candidates to be placed in the first nodes. Keeping in mind this idea the proposal is to design an automatic procedure to build the graph in any problem.

Given a problem with a set of training vectors  $\mathbf{x}_a \in \mathbb{R}^n, a = 1, \dots, l$  and a vector of labels  $\mathbf{y} \in \mathbb{R}^l, y_i \in \{i = 1, 2, \dots, N\}$ , and a set of classifiers  $A_j, j \in \{1, 2, \dots, N(N - 1)/2\}$  The basic algorithm proposed is summarized in the following steps:

1. Estimate the probabilities of each class  $C_i$  as:

$$P(C_i) = \frac{\sum_{h=1}^l u(y_h = i)}{l} . \tag{4}$$

Where  $u(\cdot)$  is the step function.

2. Calculate the number of operations associated to the class  $C_i$  as:

$$N_{opi} = \sum_{j=1}^{N-1} \text{adds}(A_{j,i}) + k_1 \text{mult}(A_{j,i}) + k_2 \text{exp}(A_{j,i}) . \tag{5}$$

Where  $k_1$  and  $k_2$  are two constants calculated as the time needed to compute a multiplication and a exponential function, taking as reference the time needed to calculate an addition.

3. Calculate the list  $L$  as following until all the classes are included:

$$L(i) = \arg \min_i (N_{opi} P(C_i)) . \tag{6}$$

4. Build the graph as shown in Fig. (4). The first classifier is the one that discriminates between  $L(1)$  and  $L(2)$ . Then, the next layer is composed by two classifiers, separating the next element of  $L$ , in the example case class 1, and the previous classes. The process is repeated until the end of the list, building in each layer as many classifiers as the number of classes have that been added in the previous layers.

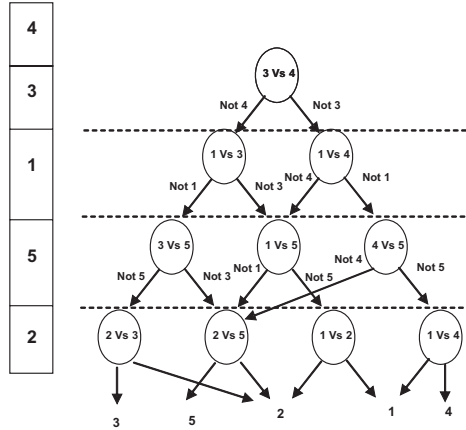


Fig. 4. Graph proposed for 4 classes with an ordered list

### 4 Results and Discussion

The proposed method, Graph Ordered SVM (GOSVM) was evaluated on different datasets obtained from the UCI [8] repository. The dataset named Cover Type was randomly reduced. We have compared these datasets using also the DAGSVM method, as described in [6] and the One-Against-Rest method coupling a probabilistic estimator (OARPSVM) as described in [4]. In order to see the performance of the optimization, the parameters of each binary classifier, the  $C$  and  $\gamma$  parameter were tuned using PSO only in the GOSVM method. DAGSVM and OARPSVM methods were trained with common  $C$  and  $\gamma$  parameters. The procedure to select these parameters was done in the classical way, that is, fixing in first place the  $C$  parameter and searching for the optimal  $\gamma$  and then searching the  $C$  parameter. In all cases, the adjustment of the parameters was done using the training set itself with a 5-crossvalidation error function. Selecting these parameters with an external test set could lead us to not generalize the problem. The results obtained are shown in Table(I). It can be appreciated how the proposed method, combining the PSO optimization for each classifier and the order of the graph give a slightly better accuracy in all cases and an important reduction in the number of operations needed in the test phase. For the Cover Type dataset the error obtained is very high, but this dataset is known to be a hard classification problem and worse results were obtained using other learning methods like neural networks. It is specially meaningful the reduction in the number of operations needed in this last case comparing to the DAGSVM and OARPSVM methods, while the accuracy achieved does not present an important improvement. This behavior can be explained due to the function (3) used in the optimization problem.

Once that both improvements have been tested, to adjust the parameters of each classifier and to order the graph, we have compared the GOSVM method

**Table 1.** Results for different datasets comparing the proposed method with the DAGSVM and one-against-all with probabilistic output

DATASET	N. of Classes	N. of Feat.	Samp. Train	Samp. Test	GOSVM		DAGSVM		OARPSVM	
					Error (%)	N° Ops ( $10^9$ )	Error (%)	N° Ops ( $10^9$ )	Error (%)	N° Ops ( $10^9$ )
SEG. IMAGE	7	18	10000	36200	0.11	6.77	0.44	16.30	0.34	59.51
WAV	3	21	6000	94000	5.04	222.93	6.10	361.0	6.06	462.34
SATELLITE	6	36	40000	88700	0	373.64	0.674	498.86	1.008	1942.5
COVER TYPE	7	54	1120	50000	39.08	1.04	39.51	139.54	40.13	371.31

with the DAGSVM, but in this case the parameters of the binary classifiers have also been optimized for each one. Results are shown in Table(2). It can be appreciated that the reduced number of operations is caused not only by the order of the graph but also by the minimization in (3). However, the order of the graph, as it has been proposed can achieve better results in the operations needed, except in the Cover Type case. This behavior has a clear explanation since the arrangement is based on the estimation of the probability of each class. In the training set of this problem, all the classes have the same probability whereas in the test set some classes have much less probability than others. It can be said that the proposal method to order the graph does not have success when the probabilities of the training set are quite different than in the test set.

**Table 2.** Results for different datasets with the parameters adjusted for each binary classifier

DATASET	GOSVM		DAGSVM	
	Error (%)	N° Ops ( $10^9$ )	Error (%)	N° Ops ( $10^9$ )
SEG. IMAGE	0.11	6.77	0.12	12.54
WAV	5.04	222.93	5.02	271.87
LETTER	2.65	9.76	2.70	11.14
SATELLITE	0	373.64	0	425.32
COVER TYPE	39.08	1.04	38.97	0.96

## 5 Conclusion

In this work we have proposed two new ideas to optimize the behavior of a multi-class SVM in a one-against-one approach. Adjusting the value of the parameters each binary classifier improves the success of classification and in this work we have exposed a method to make this tuning without searching the whole space. The order of the nodes in the graph has not great influence in the success rate of classification, but it has an important effect on the average number of operations needed in the test phase.

Model selection is still an open field to research, and in future works some other functions optimizers will be tested searching also for other kernel types. There is also an open research line to test the method here exposed to some research areas, where the number of classes is very high.

**Acknowledgments.** This work was supported by Comunidad of Madrid project CAM-UAH 2005/031.

## References

1. Vapnik, N.V.: The Nature of Statistical Learning Theory. Springer-Verlag, Berlin. (2000) 1ed: 1998.
2. Wang, L.: Support Vector Machines: Theory and Applications. Springer-Verlag, Berlin. (2005)
3. J. Weston and C. Watkins: Multi-class support vector machines. Technical report, Royal Holloway University of London (1998)
4. Platt, J.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In Smola, A., Schölkopf, B., Schuurmans, D., eds.: Advances in Large Margin Classifiers. MIT Press (1999) 61–74
5. Kreßel, U.: Pairwise classification and support vector machines. In Schölkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods – Support Vector Learning. MIT Press (1998) 225–268
6. Platt, J.: Large margin dags for multiclass classification. In Solla, S., Keen, T., Müller, K., eds.: Advances in Neural Information Processing Systems. MIT Press (2000) 547–553
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks. IEEE Press, Piscataway, NJ (1995) 1942–1948
8. D.J. Newman, S. Hettich, C.B., Merz, C.: UCI repository of machine learning databases (1998)

# Applying Dynamic Fuzzy Model in Combination with Support Vector Machine to Explore Stock Market Dynamism

Deng-Yiv Chiu and Ping-Jie Chen

Department of Information Management, ChungHua University  
Hsin-chu City, Taiwan 300, R.O.C.  
{chiuden, m09310006}@chu.edu.tw

**Abstract.** In the study, a new dynamic fuzzy model is proposed in combination with support vector machine (SVM) to explore stock market dynamism. The fuzzy model integrates various factors with influential degree as the input variables, and the genetic algorithm (GA) adjusts the influential degree of each input variable dynamically. SVM then serves to predict stock market dynamism in the next phase. In the meanwhile, the multiperiod experiment method is designed to simulate the volatility of stock market. Then, we compare it with other methods. The model from the study does generate better results than others.

## 1 Introduction

Stock market is a complicated and volatile system due to too many possible influential factors. In the past studies, as a result, dynamism in the stock market was often considered as random movement. Nevertheless, according to the researches in the recent years, it is not entirely random. Instead, it is highly complicated and volatile [1]. Many factors, including macroeconomic variables and stock market technical indicators, have been proven to have a certain level of forecast capability on stock market during a certain period of time [2]. In the past decade, various methods have been widely applied in the stock market forecast such as linear and nonlinear mathematical models or multi-agent mechanism [3] to simulate the potential stock market transaction mechanism, such as artificial neural network(ANN) of multiple layers of threshold nonlinear function. Because of the advantages of arbitrary function approximation and needless of statistics assumption, ANN is widely applied in the simulation of potential market transaction mechanism [4]. Also, to improve the forecast performance, some machine learning methods are applied. For example, genetic algorithm (GA) is used to reduce input feature dimension and select better model parameters [5] to increase the forecast accuracy rate.

Support vector machine (SVM) is a newly developed mathematical model with outstanding performances in handling high dimension entry space problems. Such a feature leads to a better performance of SVM in simulating potential market transaction mechanism than other methods.

Although the numerous related researches bring highly remarkable achievements, none of the models can continuously and successfully predict the dynamism of stock market [6]. The possible reasons are the high volatility, uncoordinate of various input variables and the selection of dynamic input variables. The volatility of the stock market makes factors influencing stock market change with time. An optimized forecast model is unable to guarantee to have the same forecast performance even after successful forecast of stock market dynamism during a certain period of time. For the selection of input variables, too few input variables will lead to inability to predict market mechanism due to insufficient factors and reduction of forecast accuracy. Too many input variables of forecast model will, however, bring too many noises and cause overfitting.

To improve the problems above, a new dynamic fuzzy model integrated with SVM is used in this study to explore the stock market dynamism. The fuzzy model, adjustable with time, is first used to consider influence factors with different features such as macroeconomic variables, stock and futures technical indicators. GA locates the optimal parameters of fuzzy model for each influence factor changing with the time. Multiperiod experiment method divides data into many sections to train the forecast model with the earlier section data to predict the latter section data in order to simulate the stock market volatility. With the newly found influential degree of input variables, SVM handles high dimension input space without causing overfitting [7] to explore the stock market movement dynamism in the next phase. Then, we compare it with other methods. The model from the study does generate better results than others.

## 2 Methodology and Architecture

In the study, we use fuzzy theory to coordinate the macroeconomic variable, stock market technical indicators and futures indexes. GA serves to dynamically adjust the fuzzy model parameters of each factor to determine the influential degree. Then, SVM is used to locate the approximate optimal parameters of fuzzy model. The integrated architecture is shown in Figure 1.

First, the parameters needed by fuzzy model for each factor are generated randomly. Then, GA adjusts the influential degrees of factors, and SVM is used to testify the new adjustment of each factor during the training period and produce accuracy rate. With the accuracy rate used as fitness function value, GA determines whether to conduct evolution or whether target is reached. In the event of evolution, after selection, crossover and mutation, parameters of new fuzzy model are generated. Otherwise, when the termination condition is reached, SVM is finally used again to forecast the stock market dynamism by employing optimal parameters.

### 2.1 Dynamic Fuzzy Model

To adjust the influential degree of input variables changing with time, we propose the dynamically adjustable fuzzy model to solve the issue. Each influential degree



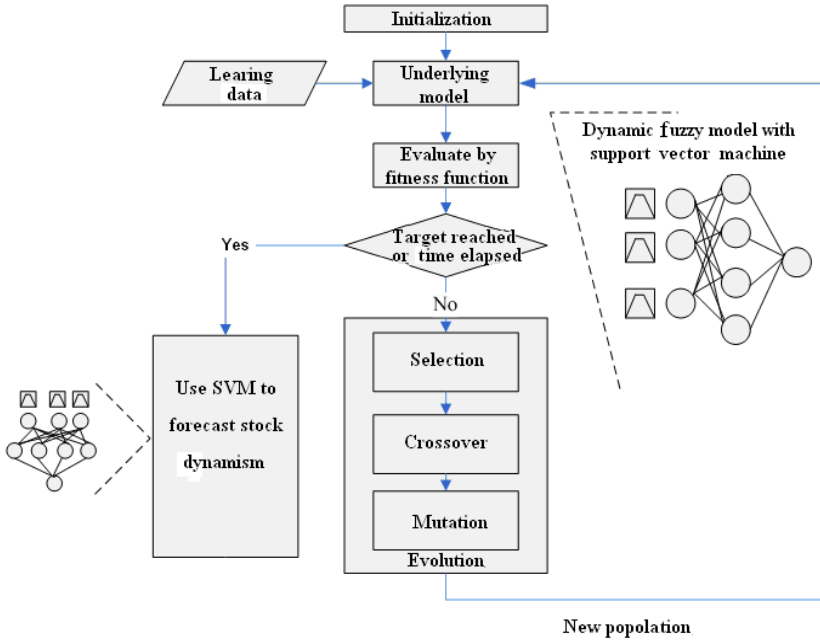


Fig. 1. The architecture of proposed integrated model

of input variable (ID) is determined by the adjusted scaled index, SI, and the influential degree of the variable changing with time,  $\mu_A(t)$ , as shown in Formula 1, where t is moment, k is an input factor.

$$ID_{k,t} = SI_{k,t} * \mu_A(t) \tag{1}$$

Due to the different field ranges of input variables, to increase the accuracy, we adopt linear transference to adjust the variable to the range of [-1, 1] as shown in Formula 2.

$$SI_{k,t} = 2 \cdot \frac{x_{k,t} - \min(x_k)}{\max(x_k) - \min(x_k)} - 1 \tag{2}$$

To reflect the changes of variables affecting the stock market with time, in the condition of considering the complexity of calculation and actual improvement of forecast accuracy, we compare the trapezoid, triangle and Gaussian membership functions and adopt the trapezoid membership function to simulate the changes. We adjust the membership function  $\mu_A(t)$  of time (t) as shown in Formula 3.

$$\mu_A(t) = \begin{cases} 0 & , t < a_1 \\ \frac{t-a_1}{a-a_1} & , a_1 \leq t \leq a \\ 1 & , a \leq t \leq b \\ \frac{b_1-t}{b_1-b} & , b < t \leq b_1 \\ 0 & , t > b_1 \end{cases} \tag{3}$$

Each influence variable has its independent fuzzy model. The shape of each fuzzy model is determined by the four parameters,  $a_1, a, b, b_1$  which will be dynamically adjusted in accordance with the fitness function from GA to properly express the influence of the variable.

## 2.2 Model Optimization with Genetic Algorithm(GA)

GA is an efficient and better search method in the broad sense. With the simulation of biological evolution phenomenon, the parameter with higher fitness function value is left. Also, with mechanisms of crossover and mutation, etc, issue of partial minimization during search is avoided and search time is shortened. Because of the stock market dynamism nature, influential degree of a factor changes with time and the model has to be dynamically adjusted. With GA, we can locate the approximate optimal solution and the better parameters of model in a certain period of time.

In the initialization stage, the chromosome in the experiment is in real-coded, because a solution is directly represented as a vector of real-parameter decision variable. Each input variable includes five elements,  $a_1, a, b, b_1$  and range. The elements  $a_1, a, b, b_1$  determine the shape of the fuzzy model, which represents the changes of the influential degree of each variable. The range represents the published cycle of the variable. Due to the difference of each problem, GA is unable to obtain the same search results with fixed detailed setup. Chromosomes are selected with highest fitness value by means of the roulette wheel. Taking time and accuracy rate into consideration, crossover in the experiment adopts two-point crossover with the probability at 0.8 and two parent chromosomes can interchange anywhere to produce two new offsprings. The two-point crossover produces better performance in our experiment. In mutation, the chromosomes are transformed into binary code and mutate with probability of 0.02 by randomly changing code from "0" to "1" and vice versa. The forecast accuracy rate in the study is the criteria to evaluate the forecast model. Therefore, design of evaluation (fitness) function is the optimization of accuracy rate produced by SVM in the training period. GA locates the approximate optimal solution. Within 100<sup>th</sup> generations with change rate below 0.02, the evolution stops. The located parameters shall serve to establish the better model for a certain period of time.

## 2.3 Support Vector Machine(SVM)

After GA determines the parameters of each fuzzy model, SVM will serve to locate the relationship among influence variables (e.g. stock market technical indicators in the stock market, macroeconomic variables and futures technical indicators in the future) and stock market dynamism.

SVM defines the input variable supposition space with linear function and introduces the learning deviation to learn the mapping between input and output. As the linear fitting machine is operated in the feature space of the kernel function for learning, when the applied field has high dimension feature space

feature, SVM shall effectively avoid overfitting problem and pose excellent learning performance [7].

The main concept is to transfer the mapping of input space kernel to high dimension feature space before re-classification. To begin, SVM selects several support vectors from the training data to represent the entire data. In this study, the issue can be expressed as:

Provided the existing training data:  $(x_1, y_1), \dots, (x_p, y_p)$ , where  $x_i \in R^n, y_i \in \{1, -1\}$ ,  $p$  is the number of data and  $n$  is the dimension of stock market influence variables. When  $y$  equals to 1, the stock market goes up; when  $y$  equals to -1, the stock market goes down. In the linear analysis, in an optimal hyperplane,  $(w \cdot x) + b = 0$  can completely separate the sample into two conditions shown as below, where  $w$  is the weight vector and  $b$  is a bias.

$$(w \cdot x) + b \geq 0 \rightarrow y_i = +1,$$

$$(w \cdot x) + b \leq 0 \rightarrow y_i = -1,$$

In the linear separation, it is a typical quadratic programming problem. Lagrange formula can be used to find the solution, where  $\alpha$  is a Lagrange multiplier.

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^p \alpha_i [y_i (w \cdot x_i + b) - 1],$$

In the linear analysis, the original problem can be considered as a dual problem. To find the optimal solution, the approach is:

$$max \quad W(\alpha) = \sum_{i=1}^p \alpha_i - \frac{1}{2} \sum_{i=1}^p \sum_{j=1}^p \alpha_i \alpha_j y_i y_j x_i^T x_j.$$

Constraint:

$$\sum_{i=1}^p \alpha_i y_i = 0, \quad \alpha_i > 0, \quad i = 1, 2, \dots, p.$$

By solving the quadratic programming, the classification formula applied to forecast the next day stock market dynamism can be obtained as shown below

$$f(x) = Sign\left(\sum_{i=1}^p y_i \alpha_i (x \cdot x_i) + y_i - w \cdot x_i\right) \tag{4}$$

Any functions that meet Mercer’s condition can be kernel functions. We adopt Radial kernel function below as kernel function of SVM [7].

$$K(s, t) = exp\left(-\frac{1}{10} \|s - t\|^2\right) \tag{5}$$

### 3 Experiment

We integrate fuzzy theory, GA and SVM to explore stock market dynamism, targeting the stock market in Taiwan. The input variables in the study includes a total of 61 variables, including technical indicators in stock market and futures market and the macroeconomic indicators in Taiwan [8]. The influence factors

for both stock and future market include On balance volume(OBV), Demand index(DI), Momentum(MTM), Relative strength index(RSI), Moving average convergence and divergence(MACD), Total amount per weighted stock price index(TAPI), Psychological line(Psy), Advance decline ratio(ADR), Williams (WMS), BIAS, Oscillator(OSC), Moving average(MA), K line(K), D line(D), Perform criteria(PC), Autoregressive(AR), Different(DIF), Consistency ratio (CR), Relative strength volume(RSV) and Exponential moving average(EMA).

Macroeconomic variables include Annual change in wholesale price index (WPC), Annual change in export price index(EPC), Annual change in industrial production index(IPC), Annual change in employees on payrolls(EMPC), Gross national production(GNP), Approved outward investment by industry(AOI), Gross domestic product(GDP), Import by key trading partners(IKT), Export by key trading partners(EKT), Long term interest rate(LT), Consumer price index(CPI), Government consumption(GC), MFGs' New Orders(MNO), Average monthly working hours(AMWH), Average monthly wages and salaries(AMWS), Bank clearings(BC), Manufacturing sales(MS), Quantum of domestic traffic (QDT), Monetary aggregate(M1B), Term architecture of interest rate(TS) and Short term interest rate(ST).

The original data of stock and futures market in Taiwan are retrieved from Taiwan Stock Exchange Corporation while macroeconomic indicators are from Ministry of Economic Affairs, R.O.C.. The historical data are for two years from January, 2003 to December, 2004 for a total of 714 pieces of data. Among which, 378 pieces go up while the rest go down.

As stock market is a complicated and volatile system, in order to express the changes of influential degree of each factor effectively, we apply two methods, multiperiod and two-period. In the multiperiod method, data of every fifty days serve as one set of training data to obtain parameters of the proposed integrated model. Forecast of the next day is made with such data as shown in Figure 2. In the two-period method, all the data are divided into two halves. One half serves as training while another half for verification.

Each indicator has its independent fuzzy model. Based on the accuracy rate during past year, GA simulates the changes of influential degree. From the experiment results, most of the dynamic fuzzy models converge after 500<sup>th</sup> generations.

Take GDP as an example. From 2004/1 to 2004/12, the adjusted dynamic fuzzy model is shown in Figure 3. The influential degree of factor GDP does not

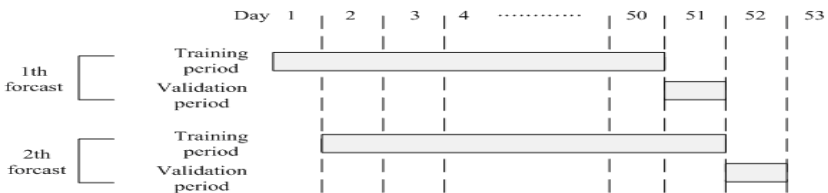
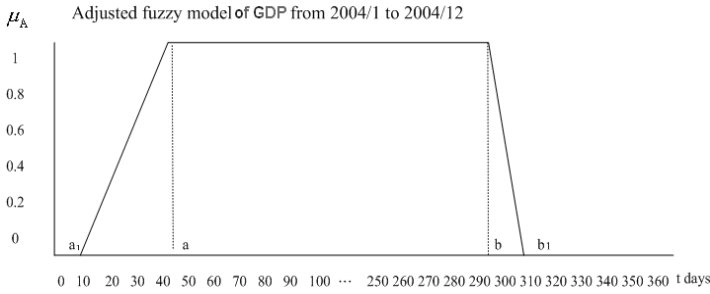


Fig. 2. Strategy for multiperiod stock market movement forecast



**Fig. 3.** Adjusted membership function for influential degree of factor GDP

start immediately. The influential degree increases gradually and reach climax in a month. Then, it lasts a long period of time. Dynamic fuzzy models located by GA differ in different time periods. This is resulted from nature of stock market changes leading to the same variable having different influence in the different periods.

Then, SVM is used to evaluate the quality of each fuzzy model. The output of SVM is accuracy rate used as value of fitness function in GA.

After termination condition in GA is reached, SVM is used again to forecast the stock dynamism. In the experiment, Radial function is adopted as kernel function.

The performance comparisons among proposed approach and other forecast methods are shown in Table 1.

**Table 1.** Performance comparisons among various approaches

Approaches	Accuracy rate(%)	
	Multi-period	Two-period
Proposed model (SVM,GA,fuzzy model)	79	73
Proposed model without GA	73	73
SVM	64.6	62.3
ANN with fuzzy model and GA	61	
ANN with fuzzy model	58.6	
Discriminant analysis	52.6	
Buy and hold	50	

The proposed model outperforms all others in the experiment. The SVM method outperforms ANN since SVM can be used to avoid the drawbacks of ANN, such as overfitting and local minimum. The models with GA outperform those without GA. It verifies that GA can dynamically adjusts the influential degree of each variable to reflect market changes and results in better performance. Also, the multiperiod approach improves the accuracy rate. This is resulted from two-period method being unable to precisely simulate market fluctuations.

Therefore, GA and multiperiod method bring higher accuracy rate. The model with fuzzy method outperforms that without fuzzy method. Without fuzzy, influential degree of each selected variable  $\mu_A(t)$  is 1. That is, influential degree of each variable remains unchanged.

## 4 Conclusions

We propose a model integrating fuzzy theory, GA and SVM to explore movements in stock market in Taiwan. The new dynamic fuzzy model not only effectively simulates market volatility but also covers influence factors of different features. The integration of high dimension variables, with features of SVM, increases the forecast accuracy rate. The integrated forecast model in this study can serve as a valuable evaluation reference for researches on internal mechanism of stock market. For future work, the interactions among various factors can be considered to improve the accuracy rate.

## References

1. Black, A.J., Mcmillan, D.G.: Non-linear Predictability of Value and Growth Stocks and Economic Activity. *Journal of Business Finance & Accounting*, Vol. 31. Blackwell Publishing. (2004) 439–474
2. Lo, A.: The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective. *Journal of Portfolio Management*, Vol. 30. New York. (2004) 15–44
3. Armano, G., Murru, A.; Roli, F.: Stock Market Prediction by A Mixture of Genetic-Neural Experts. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 16. World Scientific Publishing, Singapore. (2002) 501–526
4. Matilla-Garcia G., Arguelli, C.: A Hybrid Approach Based on Neural Networks and Genetic Algorithm to the Study of Profitability in the Spanish Stock Market. *Applied Economics Letter*, Vol. 12. Routledge part of the Taylor & Francis Group, Philadelphia. (2005) 303–308
5. Davis, L.: Genetic Algorithms and Financial Applications. In: Deboeck GJ(ed) *Trading on the Edge*. Wiley, New York. (1994) 133–147
6. Schwert, G.W.: Why Does Stock Arket Volatility Change Over Time. *The Journal of Finance*, Vol. 44. Blackwell Publishing, Oxford (1989) 1115–1167
7. Cristianini, N., Taylor, J.S.: *An Introduction to Support Vector Machines*, Cambridge University, New York (2000)
8. Dickinson, D.G.: Stock Market Integration and Macroeconomic Fundamentals: An Empirical Analysis. *Applied Financial Economics*, Vol. 10, Routledge part of the Taylor & Francis Group, Philadelphia. (2000) 261–276

# Predicting Mechanical Properties of Rubber Compounds with Neural Networks and Support Vector Machines<sup>\*</sup>

Mira Trebar and Uroš Lotrič

Faculty of Computer and Information Science,  
University of Ljubljana, Slovenia  
{mira.trebar, uros.lotric}@fri.uni-lj.si

**Abstract.** The quality of rubber compounds is assessed by rheological and mechanical tests. Since mechanical tests are very time consuming, the main idea of this work is to quest for strong nonlinear relationships between rheological and mechanical tests in order to reduce the latter. The multilayered perceptron and support vector machine combined with data preprocessing were applied to model hardness and density of the vulcanizates from the rheological parameters of the raw compounds. The results outline the advantage of proper data preprocessing.

## 1 Introduction

Rubber as the most important raw component of rubber compounds has very complex structure. Since the phenomenological models of rubber compound behavior are not suitable for practical use, the rubber producers are forced to perform numerous tests of rubber compounds in order to ascertain their quality thus assuring the quality of final rubber products.

The most important parameters that determine the quality of rubber compounds are measured by variety of rheological and mechanical tests. While the rheological tests performed on raw rubber compound asses the suitability of compounds for further production, the mechanical tests performed on vulcanizates mainly estimate the properties of final products. Due to the time consuming testing of mechanical properties, rubber producers would prefer to estimate the mechanical properties straight from rheological properties, reducing the mechanical tests to minimum.

The relations between rheological and mechanical tests are characterized by high nonlinearities emerging from the nature of both tests, first being performed at low strain and high temperature and second being performed at high strain and low temperature. The nature of the problem itself thus suggested to use neural networks and support vector machines as general nonparametric and non-linear robust models to find complex relations among both types of tests.

---

<sup>\*</sup> The work is sponsored in part by Slovenian Ministry of Education, Science and Sport by grants V2-0886, L2-6460 and L2-6143.

In the next section the applied neural network prediction model and the support vector machine model are described. In section three the rubber compound database is presented together with applied preprocessing methods. In section four the experimental setup and prediction results are given. The main findings and conclusions are outlined in the last section.

## 2 Neural Networks and Support Vector Machines

In this study two general nonparametric models, multilayer perceptron (MLP) and support vector machines (SVMs) were used to model the relationship between mechanical and rheological tests from the given input – output pairs.

A multilayer perceptron [1] is a feed-forward neural network consisting of a set of source nodes forming the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. Each computation node or a neuron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some nonlinear activation function. Layers with computation nodes can be mathematically described as  $\mathbf{y} = \mathbf{tanh}(\mathbf{\Omega}\mathbf{x} + \mathbf{\beta})$  where  $\mathbf{x}$  is a vector of inputs,  $\mathbf{y}$  is a vector of outputs,  $\mathbf{\beta}$  is a vector of biases and  $\mathbf{\Omega}$  a weight matrix. The elementwise hyperbolic tangent function introduces nonlinearity into the model. The objective of a training algorithm is to find such set of weights and biases that minimizes the performance function, defined as a squared error between calculated outputs and target values. The second-order derivative based Levenberg-Marquardt algorithm [2] was used in the training process.

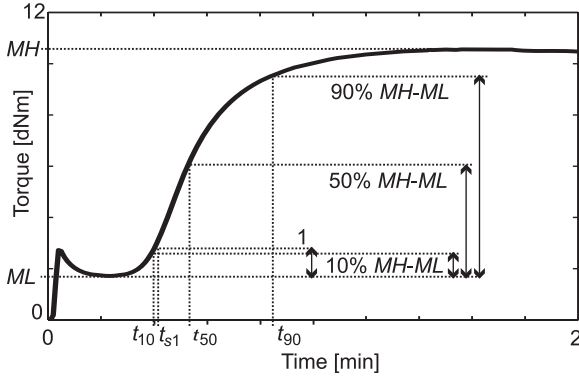
Support vector machines are general models introduced by Vapnik [3] used for solving the problems of multi dimensional function estimation [6]. The idea of the model is to find the hyperplanes that linearly separate the samples belonging to different groups up to the given precision and establish the support vectors that describe them, resulting in quadratic programming problem. To do so, the support vector machine transforms the nonlinear input data space into a higher dimensional linear feature space through a nonlinear mapping where linear function estimation over the unknown feature space is performed [4]. The transformation is performed using an inner product kernel trick, in our case, applying radial basis functions with standard deviation  $\sigma$ . To regularize the tradeoff between the model simplicity and its precision  $\epsilon$ , a regularization parameter  $U$  is used. The SVM<sup>light</sup> implementation [5] of support vector machines was used.

## 3 Rubber Compound Database

In order to estimate the quality of final rubber products, the rubber compounds are vulcanized in a laboratory and some mechanical test are performed on obtained vulcanizates. As the vulcanization phase itself, including material cooling and relaxation, lasts several hours, the rubber producers are trying to reduce the test on vulcanizates to minimum.



In order to meet the foregoing and still obtain enough information about a compound, a rheological instrument called moving dye rheometer [8] is used. In about 2 minutes it performs tests directly on raw rubber compounds under high temperature (usually above  $190^{\circ}\text{C}$ ), measuring a degree of cross-linking, manifested in torque, as a function of time during vulcanization process. From the torque curve shown in Fig. 1, some important parameters like  $ML$ ,  $MH$ ,  $t_{10}$ ,  $t_{50}$ ,  $t_{90}$  and  $t_{s1}$  are determined [7]. The result of a rheological test is actually



**Fig. 1.** Torque curve measuring the effect of cross-linking during a vulcanization process with indicated meaning of rheological parameters

a vulcanizate, unfortunately made at too high temperature. Despite of this, it is expected to obtain important information about the mechanical tests on correctly prepared vulcanizates (at temperatures around  $130^{\circ}\text{C}$ ) from the data inherently present in a torque curve.

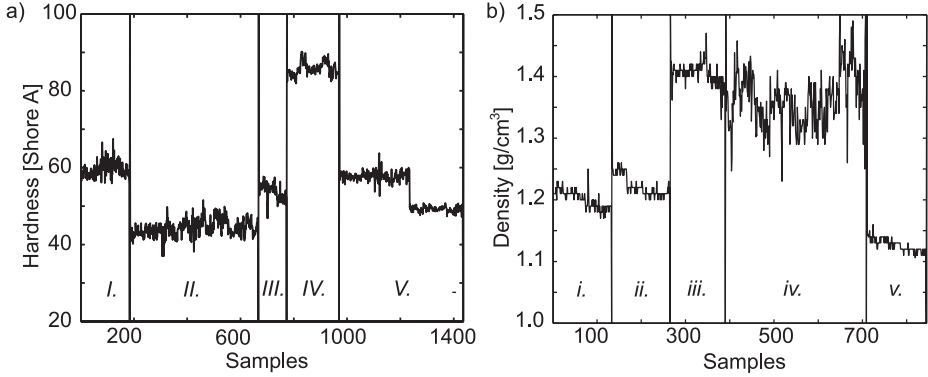
In this paper, two important parameters of rubber vulcanizates, hardness and density, are predicted from the parameters of the torque curve. The hardness (H) of vulcanizates is obtained on Shore instrument [8] by measuring the irruption of measuring pin into the rubber on scale from 0 to 100 with higher value meaning harder rubber. The density (D) of vulcanizate is calculated from the measurement of the vulcanizate weight in the air and in the water on appropriate scale.

Rubber compounds used in experiments can be divided into ten groups based on the chemical structure of rubber: natural rubber (NR), ethylene-propylene-diene (EPDM), butyl (IIR), styrene-butadiene (SBR), butadiene-acrylonitrile (NBR), polychloroprene (CR), polybutadiene (BR), polyethylene (PE), synthetic polyisoprene (IR), chlorinated butyl (CIIR) and into two groups regarding the type of active filler: carbon black (CB), silica (SI).

## 4 Preprocessing

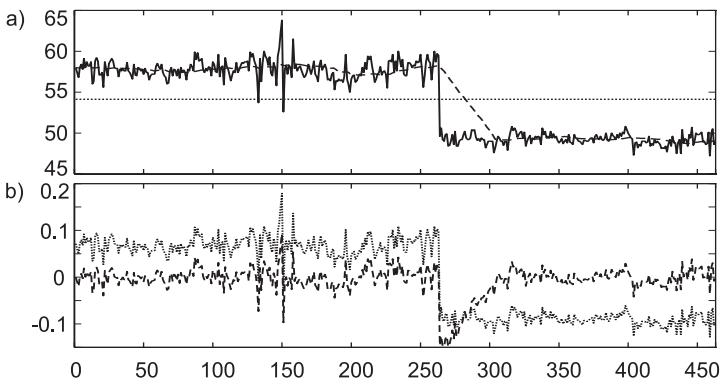
The inconsistent data and the measurements on rarely mixed compound, having less than 100 measurement in 6 years, were removed from further investigation.

Thus for hardness modeling 83 different compounds with 23989 measurements remained in the database, while for density modeling 68 compounds with 14744 samples were kept. Variation of hardness and density for five different compounds is presented in Fig. 2.



**Fig. 2.** Variation of a) hardness and b) density of five different compounds. Samples of each compound are presented in chronological order.

The purpose of laboratory testing is to find out whether a given raw compound is suitable for further production or not. Thus, instead of predicting the exact values of hardness and density, it is sufficient to determine deviation from a desired, usually average, value. This facts led us to an idea of representing each rheological and mechanical parameter with two values: an average and a relative deviation, defined as quotient between the deviation from the average and the average itself.



**Fig. 3.** a) Variation of hardness of one of the compounds used in the experiments (*continuous line*) with indicated compound average (*dotted line*) and the moving average (*dashed line*). b) Relative deviations from both averages shown at the bottom are marked correspondingly.

Two averaging methods were considered in the following experiments: (i) averaging of all values in compound and (ii) moving average with a large window covering one tenth of the compound samples. Both averaging methods are presented in Fig. 3. The latter method was introduced to compensate for the changes of compound properties usually caused by producer replacing one of the ingredients with its equivalent from different supplier.

In the following, preprocessing methods and additional features are labeled as follows: (B) basic data set with 7 rheological parameters of the torque curve on input, (AVc) basic data set with compound average preprocessing having  $2 \times 7$  inputs, (AVm), basic data set with moving average preprocessing having  $2 \times 7$  inputs. The prefix (C+) before the label indicates the usage of 12 additional inputs determining chemical structure of rubber and the type of the active filler.

## 5 Experiments

For each data set the input-output pairs were divided into the training set with 80% of input-output pairs and test set with remaining 20% of input-output pairs. The input-output pairs in test set were only used to assess performance of the models. Only the first 80% of pairs in training set were used to build the models, while the last 20% were used to reduce the effect of overfitting in case of neural networks and to set up the parameters  $\sigma$ ,  $U$  and  $\epsilon$  of the support vector machines.

In the case of multilayered perceptron models the number of nodes in on hidden layer was altered from 7 to  $2n$ , where  $n$  indicates the number of nodes in the input layer. In the case of the support vector machine models parameter  $\sigma$  was altered in the range from 0.1 to 16,  $U$  from 1 to 30 and  $\epsilon$  from 0.01 to 0.5.

Three criteria were used to evaluate the models: (i) the root mean squared error, normalized to the standard deviation (NRMSE), (ii) the mean absolute percentage error (MAPE) and (iii) the percentage of correctly classified samples (%OK). Each measurement and/or predicted result can fall into one of three classes: inside, above or below the specified limit range. A given sample is correctly classified when the measurement and the prediction fall in the same class.

### 5.1 Hardness

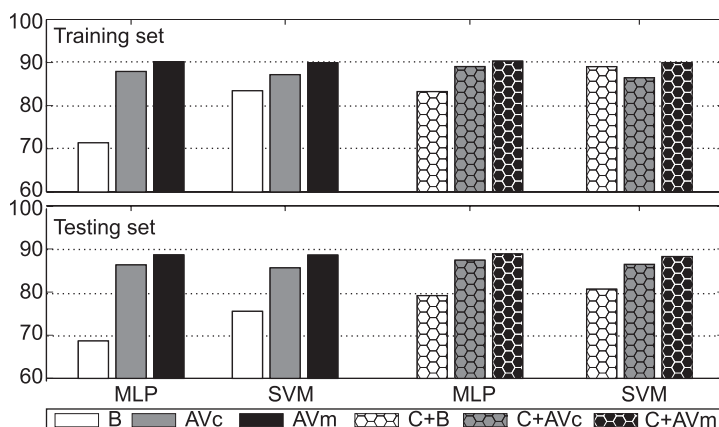
The comparison of models on the hardness data set is given in Table 1. The multilayered perceptron is presented as a triple  $n - h - 1$  where  $n$  is the number of inputs, and  $h$  number of nodes in the hidden layer. Similarly the support vector machine models are given with four parameters  $n, (U, \sigma, \epsilon)$ , representing number of inputs, regularization parameter, standard deviation, and precision, respectively. Experiments on data sets with no information about the chemical structure of compounds show that both averaging preprocessing methods considerably help the models to extract the knowledge present in data sets. Considering

**Table 1.** Prediction results for hardness data set

Data set	Model	Parameters	NRMSE		MAPE		% OK	
			train	test	train	test	train	test
B		7-7-1	0.29	0.32	0.047	0.049	71.37	68.67
AV	MLP	14-15-1	0.19	0.22	0.028	0.031	87.93	86.30
MAV		14-7-1	0.16	0.22	0.023	0.030	90.19	88.63
B		7, (20, 0.5, 0.1)	0.25	0.34	0.033	0.049	83.46	75.54
AVc	SVM	14, (10, 4.0, 0.1)	0.18	0.25	0.029	0.037	87.19	85.63
AVm		14, (1, 9.0, 0.1)	0.17	0.24	0.025	0.031	89.96	88.59
C+B		19-17-1	0.22	0.26	0.033	0.039	83.26	79.19
C+AVc	MLP	26-27-1	0.17	0.21	0.026	0.032	89.07	87.40
C+AVm		26-21-1	0.15	0.21	0.025	0.029	90.33	88.86
C+B		19, (20, 1, 0.1)	0.18	0.29	0.025	0.042	89.03	80.67
C+AVc	SVM	26, (10, 2, 0.1)	0.19	0.24	0.030	0.037	86.47	86.42
C+AVm		26, (1, 4, 0.1)	0.17	0.23	0.025	0.031	89.98	88.26

the NRMSE and the MAPE criteria the improvement is around 30% in the case of the MLP model and about 15% in the case of the SVM model. Information about the chemical structure of rubber compounds considerably improves the models of basic data set, while this advantage completely fades out with introduction of preprocessing methods. Similarly, the introduction of preprocessing methods diminishes the advantage of SVM models over MLP model on the basic data sets.

Figure 4 shows a high percentage of correctly classified measurements on training and test sets. As already noted, the preprocessing techniques help the models to overcome the lack of information about the rubber compounds chemical structure.

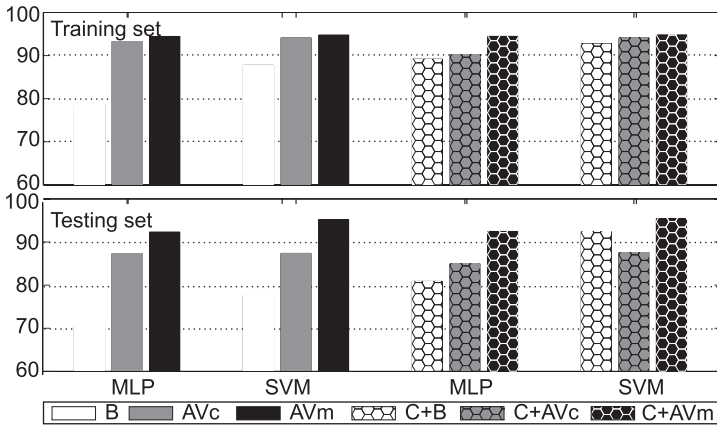
**Fig. 4.** Percentage of correctly classified measurements of hardness

### 5.2 Density

Considering Table 2, similar conclusions regarding the preprocessing can be drawn for density database. The effect of preprocessing being even more pronounced in this case is reflected in the NRMSE and the MAPE criteria becoming at least 50% better. Besides, it can be observed that the SVM models outperform the MLP models. The graphs in Fig. 5 also exhibit higher percentage of correctly classified measurements comparing to the hardness database.

**Table 2.** Prediction results for density data set

Data set	Model	Parameters	NRMSE		MAPE		% OK	
			train	test	train	test	train	test
B		7-17-1	0.44	0.64	0.019	0.028	78.59	70.27
AVc	MLP	14-9-1	0.18	0.30	0.007	0.011	93.25	87.28
AVm		14-7-1	0.15	0.23	0.005	0.008	94.43	92.31
B		7, (1, 1, 0.01)	0.41	0.77	0.014	0.031	87.73	77.55
AVc	SVM	14,(1, 25, 0.01)	0.13	0.33	0.006	0.011	94.06	87.35
AVm		14,(1, 25, 0.01)	0.11	0.22	0.005	0.007	94.77	95.18
C+B		19-27-1	0.24	0.39	0.010	0.016	89.15	80.84
C+AVc	MLP	26-7-1	0.21	0.30	0.008	0.012	90.19	84.91
C+AVm		26-21-1	0.15	0.23	0.005	0.008	94,45	92.44
C+B		19, (1, 1, 0.01)	0.19	0.48	0.007	0.020	92.71	82.37
C+AVc	SVM	26, (1, 16, 0.01)	0.13	0.34	0.006	0.012	94.06	87.52
C+AVm		26, (1, 25, 0.01)	0.11	0.22	0.004	0.007	94.78	95.38



**Fig. 5.** Percentage of correctly classified density measurements

## 6 Conclusions

Modeling of the mechanical parameters of rubber compounds from their rheological properties is burdened by high nonlinearities. Namely, the rheological

tests are performed at low strain and high temperature while the mechanical parameters are obtained in tests at high strain and low temperature.

In the paper, two models were compared, the multilayered perceptron (MLP) and the support vector machine (SVM). Both models were applied to the basic data set, two preprocessed data sets and data sets with additional information of chemical structure.

For the basic data set the SVM model outperforms the MLP model in predicting mechanical properties due to its ability of handling very high nonlinearities. Two types of preprocessing, i.e. compound average and moving average preprocessing, were used to outline the advantage of proper data preparation. Both improve the prediction results and also the classification of predicted results.

Moreover, the model with moving average preprocessing was able to extract all necessary information from the data set itself, and the additional information on chemical structure did not result in further improvement. The mechanical properties are mainly determined by the rubber bond reinforcing effect which is very similar for all rubbers and surface active fillers used in experiments.

Two modifications of the model might improve the results. Firstly, additional information on the chemical structure could be used - not only presence (0,1), but also the quantity of chemical ingredients. Secondly, some additional parameters should be retrieved from the last part of momentum curves, where the properties of rubber compound are close to the properties of vulcanizates used in mechanical tests.

## References

1. Haykin, S.: *Neural networks: a comprehensive foundation*, 2nd ed., Prentice-Hall, New Jersey (1999)
2. Hagan, M. T., Menhaj, M. B.: Training feedforward networks with the marquardt algorithm. *IEEE Trans. Neural Networks* 5(6) (1994) 989–993
3. Vapnik, V. N.: *The Nature of Statistical Learning Theory*. New York, Springer-Verlag (2000)
4. Kecman, V.: *Learning and Soft Computing*. MIT Press London (2001)
5. Joachims, T.: *SVMlight Support Vector Machine*. <http://www.cs.cornell.edu/People/tj/svm-light> (2004)
6. Vong, C. M., Wong, P. K., Li, Y. P.: Prediction of automotive engine power and torque using least squares support vector machines and Bayesian inference. *Engineering Applications of Artificial Intelligence* 19 (2006) 277–287
7. Painter, P. C., Coleman, M. M.: *Fundamentals of Polymer Science*. Technomic, Lancaster (1997)
8. Trebar, M., Lotrič, U.: Predictive data mining on rubber compound database. In: Ribeiro B., et.al. (eds.): *Adaptive and natural computing algorithms: proceedings of the International Conference in Coimbra, Portugal*, Springer, Wien New York (2005) 108-111.

# An Evolutionary Programming Based SVM Ensemble Model for Corporate Failure Prediction

Lean Yu<sup>1,2</sup>, Kin Keung Lai<sup>2,3</sup>, and Shouyang Wang<sup>1,2</sup>

<sup>1</sup> Institute of Systems Science, Academy of Mathematics and Systems Science,  
Chinese Academy of Sciences, Beijing 100080, China  
{yulean, sywang}@amss.ac.cn

<sup>2</sup> College of Business Administration, Hunan University, Changsha 410082, China

<sup>3</sup> Department of Management Sciences, City University of Hong Kong,  
Tat Chee Avenue, Kowloon, Hong Kong  
{msyulean, mskklai}@cityu.edu.hk

**Abstract.** In this study, a multistage evolutionary programming (EP) based support vector machine (SVM) ensemble model is proposed for designing a corporate bankruptcy prediction system to discriminate healthful firms from bad ones. In the proposed model, a bagging sampling technique is first used to generate different training sets. Based on the different training sets, some different SVM models with different parameters are then trained to formulate different classifiers. Finally, these different SVM classifiers are aggregated into an ensemble output using an EP approach. For illustration, the proposed SVM ensemble model is applied to a real-world corporate failure prediction problem.

## 1 Introduction

Ensemble learning has been turned out to be an efficient way to achieve high prediction/classification performance, especially in fields where the development of a powerful single classifier system requires considerable efforts [1]. According to Olmeda and Fernandez [2], an optimal system may not be an individual model but the combination of several of them from a decision support system (DSS) perspective. Usually, ensemble model outperforms the individual models, whose performance is limited by the imperfection of feature extraction, learning/classification algorithms, and the inadequacy of training data. Another reason supporting this argument is that different individual models have their inherent drawbacks and thus aggregating them may lead to a good classifier with high generalization capability. From the above descriptions, we can conclude that there are two essential requirements to the ensemble members and the ensemble strategy. The first is that the ensemble members must be diverse or complementary, i.e., classifiers must show different classification properties. Another condition is that an optimal ensemble strategy is also required to fuse a set of complementary classifiers [1].

To achieve high performance, this study utilizes a new machine learning tool — support vector machine (SVM) first proposed by Vapnik [3] — as a generic model for ensemble learning. The main reasons of selecting SVM as ensemble learning tool reflect the following aspects. First of all, SVM requires less prior assumptions about the input data, such as normal distribution and continuousness, different from statistical

models. Second, they can perform a nonlinear mapping from an original input space into a high dimensional feature space, in which it constructs a linear discriminant function to replace the nonlinear function in the original low dimension input space. This character also solves the dimension disaster problem because its computational complexity is not dependent on the sample dimension. Third, they attempt to learn the separating hyperplane to maximize the margin, therefore implementing structural risk minimization and realizing good generalization ability. This pattern can directly help SVM escape local minima and avoid overfitting problem, which are often shown in the training of artificial neural networks (ANN) [3]. These important characteristics will also make SVM popular in many practical applications.

The basic procedure of using the SVM to construct an ensemble classifier consists of three stages. In the first stage, an initial dataset is transformed into some different training sets by certain sampling algorithms. In this study, a bagging sampling approach [4] is used to generate different training datasets. In the second stage, the SVM models are trained by various training datasets from the previous stage to formulate some generic classifiers with different classification properties. Because different training datasets have different information, the generic classifiers produced by these different datasets should be diverse in terms of some previous empirical analysis [1, 5-6]. In the final stage, these different SVM classifiers are aggregated into an ensemble output using an integration approach. In this study, we use classification accuracy maximization principle to construct an optimal ensemble classifier. Particularly, an evolutionary programming (EP) algorithm [7] is used to solve the maximization problem. For testing purpose, a real-world corporate bankruptcy dataset are used to verify the effectiveness of the proposed SVM ensemble model.

The main motivation of this study is to design a high-performance classifier for corporate failure prediction and compare its performance with other existing approaches. The rest of the study is organized as follows. The next section presents a formulation process of a multistage SVM ensemble model in detail. For illustration and verification purposes, a practical experiment is performed and corresponding results are reported in Section 3. And Section 4 concludes the study.

## 2 Methodology Formulation Process

In this section, a triple-stage SVM ensemble model is proposed for classification. The basic idea of SVM ensemble originated from using all the valuable information hidden in all individual SVM classifiers, where each can contribute to the improvement of generalization. In our proposed SVM ensemble model, a bagging sampling approach is first used to generate different training sets for guaranteeing enough training data. Using these different training datasets, multiple individual SVM classifiers can be then formulated as ensemble members or components. Finally, all ensemble members are aggregated into an ensemble output.

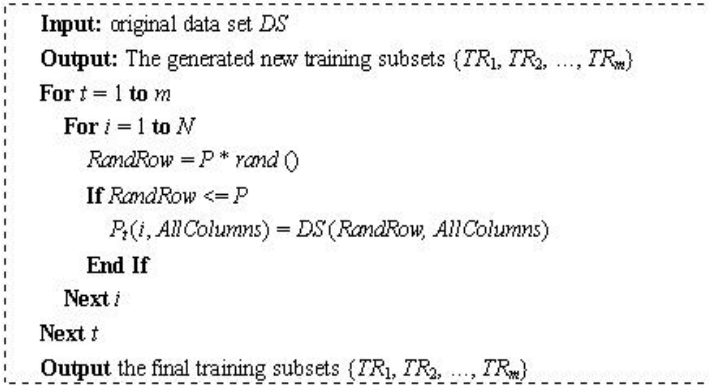
### 2.1 Stage I: Data Sampling

Data sampling is one of the most important steps in designing an ensemble model. This step is necessary and crucial for many reasons, most importantly to determine if



the training set at hand is functionally or structurally divisible into some distinct training sets. In this study, the bootstrap aggregating (**bagging**) proposed by Breiman [4] is utilized as data sampling tool.

Bagging is a widely used data sampling method in the machine learning. Given that the size of the original data set  $DS$  is  $P$ , the size of new training data is  $N$ , and the number of new training data items is  $m$ , the bagging algorithm of generate new training subsets can be shown in Fig. 1.



**Fig. 1.** The bagging algorithm

The bagging algorithm is very efficient in constructing a reasonable size of training set when the size of the original data set is small due to the feature of its random sampling with replacement. Therefore, bagging is a useful data sampling method for machine learning. In this study, we use the bagging to generate different training sets.

## 2.2 Stage II: Individual SVM Classifiers Creation

According to the definition of effective ensemble classifiers by Hansen and Salamon [8], ‘a necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse.’ That is, an effective ensemble classifier consisting of diverse models with much disagreement is more likely to have a good generalization performance. Therefore, how to generate the diverse model is a crucial factor. For the SVM model, several methods can be used to create different ensemble members. Such methods basically rely on varying the parameters related to the design and to the training of SVM models. In particular, some main ways include the following aspects:

(i) Using different kernel functions in SVM model. For example, polynomial function and Gaussian function are typical kernel functions.

(ii) Changing the SVM model parameters. Typically, margin parameter  $C$  and kernel parameter  $\sigma^2$  are usually considered.

(iii) Varying training data sets. Because different datasets contains different information, different datasets can generate diverse model with different model parameters.

In our study, the third way is selected because the previous phase creates many different training datasets. With these different training datasets, diverse SVM classifiers with disagreement as ensemble members can be generated. Interested readers can be referred to Vapnik [3] for more details about SVM classification.

### 2.3 Stage III: Ensemble Members Aggregation

When individual SVM classifiers are generated, each classifier can output its own results in terms of testing set. Before integrating these ensemble members, strategies of selecting ensemble members must be noted. Generally, these strategies can be divided into two categories: (i) generating an exact number of ensemble members; and (ii) overproducing ensemble members and then selected a subset of these [9].

For the first strategy, several common ensemble approaches, e.g., boosting [10], can be employed to generate the exact number of diverse ensemble members for integration purpose. Therefore, no selection process will be used and all generated ensemble members will be combined into an aggregated output. For the second strategy, its main aim is to create a large set of ensemble candidates and then choose some most diverse members for integration. The selection criterion is some error diversity measures, which is introduced in detail by Partridge and Yates [11]. Because the first strategy is based upon the idea of creating diverse neural networks at the early stage of design, it is better than the second one, especially for some situations where access to powerful computing resources is restricted. The main reason is that the second strategy cannot avoid occupying much computing time and storage while creating a large number of ensemble candidates, some of which are to be later discarded.

Actually, a simple way to take into account different opinions is to take the vote of the majority of the population of classifiers. In the existing literature, majority voting is the most widely used ensemble strategy for classification problems due to its easy implementation. Ensemble members' voting determines the final decision. Usually, it takes over half the ensemble to agree a result for it to be accepted as the final output of the ensemble regardless of the diversity and accuracy of each model's generalization. However, majority voting has several important shortcomings. First of all, it ignores the fact some classifiers that lie in a minority sometimes do produce the correct results. Second, if too many inefficient and uncorrelated classifiers are considered, the vote of the majority would lead to worse prediction than the ones obtained by using a single classifier. Third, it does not consider for their different expected performance when they are employed in particular circumstances, such as plausibility of outliers. At the stage of integration, it ignores the existence of diversity that is the motivation for ensembles. Finally, this method can not be used when the classes are continuous [2, 9]. For these reasons, an additive method that permits a continuous aggregation of predictions should be preferred. In this study, we propose an evolutionary programming (EP) [7] based approach to realize the classification/prediction accuracy maximization. The main reason of selecting EP rather than genetic algorithm (GA) is its ability to work with continuous parameters rather than binary coded independent variables. This makes the implementation of method easier and more accurate. Moreover, because of the self-adaptation mechanism in EP, global convergence is achieved faster compared to GA [12].

Suppose that we create  $p$  classifiers and let  $c_{ij}$  be the classification results that classifier  $j, j=1, 2, \dots, p$  makes of sample  $i, i=1, 2, \dots, N$ . Without loss of generality, we assume there are only two classes (failed and non-failed firms) in the data samples, i.e.,  $c_{ij} \in \{0,1\}$  for all  $i, j$ . Let  $C_i^w = \text{Sign}(\sum_{j=1}^p w_j c_{ij} - \theta)$  be the ensemble prediction of the data sample  $i$ , where  $w_j$  is the weight assigned to classifier  $j$ ,  $\theta$  is a confidence threshold and  $\text{sign}(\cdot)$  is a sign function. For corporate failure prediction problem, an analyst can adjust the confidence threshold  $\theta$  to change the final classification results. Only when the ensemble output is larger than the cutoff, the firm can be classified as good or healthful firm. Let  $A_i(w)$  be the associated accuracy of classification:

$$A_i(w) = \begin{cases} a_1 & \text{if } C_i^w = 0 \text{ and } C_i^s = 0, \\ a_2 & \text{if } C_i^w = 1 \text{ and } C_i^s = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where  $C_i^w$  is the classification result of the ensemble classifier,  $C_i^s$  is the actual observed class of data sample itself,  $a_1$  and  $a_2$  are the Type I and Type II accuracy, respectively, whose definitions can be referred to Lai et al. [1, 5-6].

The current problem is how to formulate an optimal combination of classifiers for ensemble prediction. A natural idea is to find the optimal combination of weights  $w^* = (w_1^*, w_2^*, \dots, w_p^*)$  by maximizing total classification accuracy including Type I and II accuracy. Usually, the classification accuracy can be estimated through  $k$ -fold cross-validation (CV) technique. With the principle of total classification accuracy maximization, the above problem can be summarized as an optimization problem:

$$(P) \begin{cases} \max_w A(w) = \sum_{i=1}^M A_i(w) \\ \text{s.t. } C_i^w = \text{sign}(\sum_{j=1}^p w_j c_{ij} - \theta), i=1,2,\dots,M \\ A_i(w) = \begin{cases} a_1 & \text{if } C_i^w = 0 \text{ and } C_i^s = 0, \\ a_2 & \text{if } C_i^w = 1 \text{ and } C_i^s = 1, \\ 0 & \text{otherwise.} \end{cases} \end{cases} \tag{2}$$

where  $M$  is the size of cross-validation set and other symbols are similar to the above notations.

Since the constraint  $C_i^w$  is a nonlinear threshold function and the  $A_i(w)$  is a step function, the optimization methods assuming differentiability of the objective function may have some problems. Therefore the above problem cannot be solved with classical optimization methods. For this reason, an EP algorithm [7] is proposed to solve the optimization problem indicated in (2) because EP is a useful method of optimization when other techniques such as gradient descent or direct analytical method are impossible. For the above problem, the EP is described as follows:

(i) Create an initial set of  $L$  solution vectors  $w_r = (w_{r1}, w_{r2}, \dots, w_{rp}), r=1,2,\dots,L$  for above optimization problems by randomly sampling the interval  $[x, y], x, y \in R$ . Each population or individual  $w_r$  can be seen as a trial solution.

(ii) Evaluate the objective function of each of the vectors  $A(w_r)$ . Here  $A(w_r)$  is called as the fitness of  $w_r$ .

(iii) Add a multivariate Gaussian vector  $\Delta_r = N(0, G(A(w_r)))$  to the vector  $w_r$  to obtain  $w'_r = w_r + \Delta_r$ , where  $G$  is an appropriate monotone function. Re-evaluate  $A(w'_r)$ . Here  $G(A(w_r))$  is called as mutation rate and  $w'_r$  is called as an offspring of individual  $w_r$ .

(iv) Define  $\bar{w}_i = w_i, \bar{w}_{i+L} = w'_i, i = 1, 2, \dots, L, \bar{C} = \bar{w}_i, i = 1, 2, \dots, 2L$ . For every  $\bar{w}_j, j = 1, 2, \dots, 2L$ , choose  $q$  vectors  $\bar{w}_*$  from  $\bar{C}$  at random. If  $A(\bar{w}_j) > A(\bar{w}_*)$ , assign  $\bar{w}_j$  as a “winner”.

(v) Choose the  $L$  individuals with more number of “winners”  $w_i^*, i = 1, 2, \dots, L$ . If the stop criteria are not fulfilled, let  $w_r = w_i^*, i = 1, 2, \dots, L$ , generation = generation + 1 and go to step 2.

Using this EP algorithm, an optimal combination,  $w^*$ , of classifiers that maximizes the total classification accuracy is formulated. To verify the effectiveness of the proposed optimal ensemble classifier, a real-world corporate failure dataset is used.

### 3 Experiment Analysis

The data used in this study is about UK firms from the Financial Analysis Made Easy (FAME) database which can be found in the Appendix of [13]. It contains 30 failed and 30 non-failed firms. 12 variables are used as the firms’ characteristics description:

- (1) Sales;
- (2) ROCE: profit before tax/capital employed (%);
- (3) FFTL: funds flow (earnings before interest, tax & depreciation)/total liabilities;
- (4) GEAR: (current liabilities + long-term debt)/total assets;
- (5) CLTA: current liabilities/total assets;
- (6) CACL: current assets/current liabilities;
- (7) QACL: (current assets)/current liabilities;
- (8) WCTA: (current assets – current liabilities)/total assets;
- (9) LAG: number of days between account year end and the date the annual report and accounts were failed at company registry;
- (10) AGE: number of years the firm has been operating since incorporation date;
- (11) CHAUD: coded 1 if changed auditor in previous three years, 0 otherwise;
- (12) BIG6: coded 1 if company auditor is a Big6 auditor, 0 otherwise.

This study is to identify the two classes of corporate bankruptcy problem: failed and non-failed. They are categorized as “0” or “1” in the research data. “0” means failed firm and “1” represents non-failed one. In this empirical test, 40 firms are randomly drawn as the training sample. Due to the scarcity of data, we make the number of good firms equal to the number of bad firms in both the training and testing samples, so as to avoid the embarrassing situations that just two or three good (or bad, equally likely) firms in the testing sample. Thus the training sample includes 20 data of each class. Its aim is to minimize the effect of such factors as industry or size that in some cases can be very important. For the training sample, we do a fifteen-fold

cross validation (i.e,  $k=15$ ) experiments to determine the best single model. Except from the above training sample, the testing sample was collected using a similar approach. The testing sample consists of 10 failed and 10 non-failed firms. The testing data is used to test results with the data that is not utilized to develop the model. The prediction performance is evaluated by the Type I accuracy, Type II accuracy and total accuracy [1, 5-6].

For constructing an EP-based SVM ensemble model, 20 training sets are generated by bagging algorithm. For each ensemble member, the kernel function is Gaussian function. Related parameters of Gaussian function are obtained by trail and error. In the process of integration, the initial solution vector is between zero and one. For training, the individual size is set to 100 and number of runs is 500. The mutation rate is determined by the Gaussian function, as shown in the previous section. Meantime, the study compares the prediction performance with several commonly used models, such as linear discriminant analysis (LDA), logit regression analysis (LogR), artificial neural network (ANN) and single SVM model. For the ANN models, a three-layer back-propagation neural network (BPNN) with 25 TANSIG neurons in the hidden layer and one PURELIN neuron in the output layer is used. The network training function is the TRAINLM. Besides, the learning rate and momentum rate is set to 0.15 and 0.25. The accepted average squared error is 0.005 and the training epochs are 2000. In the single SVM, the kernel function is Gaussian function with regularization parameter  $C = 40$  and  $\sigma^2=10$ . The above parameters of ANN and SVM are obtained by trial and error using cross-validation techniques.

The all classification results are reported in Table 1. Note that the results reported in Table 1 are the average of fifteen-fold cross-validation experiments and the values in bracket are standard deviations of fifteen-fold cross-validation experiments

**Table 1.** The results of SVM ensemble and its comparisons with other classifiers

Method	Type I (%)	Type II (%)	Overall (%)
LDA	67.67 [9.23]	71.33 [7.67]	69.50 [8.55]
LogR	72.67 [6.78]	75.00 [7.56]	73.83 [7.15]
ANN	70.67 [8.16]	74.33 [7.26]	72.67 [7.73]
SVM	77.00 [4.14]	82.67 [6.51]	79.83 [6.09]
SVM ensemble	81.33 [4.42]	88.33 [5.56]	84.83 [6.09]

As can be seen from Table 1, we can find the following several conclusions:

(1) The SVM ensemble model is the best of all the listed models in terms of Type I accuracy and Type II accuracy as well as total accuracy, indicating that the proposed evolutionary programming based SVM ensemble model is a promising technique for corporate failure prediction.

(2) For three evaluation criteria, the EP-based SVM ensemble model performs the best, followed by the single SVM, logistics regression, ANN model, and linear discriminant analysis model. Interestingly, the performance of logistic regression model

is better than that of ANN model. The possible reason leading to this conclusion may be data scarcity or other unknown reasons.

(3) Although the performance of the ANN model is worse than that of the logit regression for Type II accuracy, the robustness of logit regression is slightly worse than that of ANN model. The reasons are worth exploring further in the near future.

(4) Using two tailed *t*-test, we find that the differences among the former three methods are insignificant at 5% significance level, and there are significant differences between the former three methods and the latter two methods at 1% significance level. Furthermore, there is a significant difference between the single SVM method and the SVM ensemble model at 10% significance level. From the general view, the EP-based SVM ensemble dominates the other four classifiers, revealing the proposed EP-based SVM ensemble is an effective tool for corporate failure prediction.

## 4 Conclusions

In this study, a novel evolutionary programming (EP) based support vector machine ensemble classification method is proposed for corporate failure prediction. Through the practical data experiment, we have obtained good classification results and meantime demonstrated that the SVM ensemble model outperforms all the benchmark models listed in this study. These advantages imply that the novel SVM ensemble technique can provide a feasible solution to corporate bankruptcy prediction problem.

**Acknowledgements.** This work is supported by the grants from the National Natural Science Foundation of China (NSFC No. 70601029), the Chinese Academy of Sciences (CAS No. 3547600), the Academy of Mathematics and Systems Sciences (AMSS No. 3543500) of CAS, and the Strategic Research Grant of City University of Hong Kong (SRG No. 7001806).

## References

1. Lai, K.K., Yu, L., Wang, S.Y., Zhou, L.G.: Credit Risk Analysis Using a Reliability-based Neural Network Ensemble Model. *Lecture Notes in Computer Science* 4132 (2006) 682-690
2. Olmeda, I., Fernandez, E.: Hybrid Classifiers for Financial Multicriteria Decision Making: The Case of Bankruptcy Prediction. *Computational Economics* 10 (1997) 317-335
3. Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer, New York (1995)
4. Breiman, L.: Bagging Predictors. *Machine Learning* 26 (1996) 123-140
5. Lai, K.K., Yu, L., Wang, S.Y., Zhou, L.G.: Neural Network Meta-learning for Credit Scoring. *Lecture Notes in Computer Science* 4113 (2006) 403-408
6. Lai, K.K., Yu, L., Huang, W., Wang, S.Y.: A Novel Support Vector Machine Metamodel for Business Risk Identification. *Lecture Notes in Artificial Intelligence* 4099 (2006) 480-484
7. Fogel, D.B.: *System Identification through Simulated Evolution: A Machine Learning Approach to Modeling*. Ginn Press, Needham, MA (1991)
8. Hansen, L.K., Salamon, P.: Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (1990) 993-1001

9. Yang, S., Browne, A.: Neural Network Ensembles: Combining Multiple Models for Enhanced Performance Using a Multistage Approach. *Expert Systems* 21 (2004) 279-288
10. Schapire, R.E.: The Strength of Weak Learnability. *Machine Learning* 5 (1990) 197-227
11. Partridge, D., Yates, W.B.: Engineering Multiversion Neural-Net Systems. *Neural Computation* 8 (1996) 869-893
12. Damavandi, N., Safavi-Naeini, S.: A Robust Model Parameter Extraction Technique Based on Meta-Evolutionary Programming for High Speed/High Frequency Package Interconnects. 2001 Canadian Conference on Electrical and Computer Engineering - IEEE-CCECE, Toronto, Ontario, Canada (2001) 1151-1155
13. Beynon, M.J., Peel, M.J.: Variable Precision Rough Set Theory and Data Discretisation: An Application to Corporate Failure Prediction. *Omega* 29 (2001) 561-576

# Novel Multi-layer Non-negative Tensor Factorization with Sparsity Constraints\*

Andrzej Cichocki<sup>1,\*\*</sup>, Rafal Zdunek<sup>1,\*\*\*</sup>, Seungjin Choi<sup>2</sup>,  
Robert Plemmons<sup>3</sup>, and Shun-ichi Amari<sup>1</sup>

<sup>1</sup> RIKEN Brain Science Institute, Wako-shi, Japan  
a.cichocki@riken.jp

<http://www.bsp.brain.riken.jp>  
<sup>2</sup> POSTECH, Korea

<http://www.postech.ac.kr/~seungjin>

<sup>3</sup> Dept. of Mathematics and Computer Science, Wake Forest University, USA  
plemmons@wfu.edu  
<http://www.wfu.edu/~plemmons>

**Abstract.** In this paper we present a new method of 3D non-negative tensor factorization (NTF) that is robust in the presence of noise and has many potential applications, including multi-way blind source separation (BSS), multi-sensory or multi-dimensional data analysis, and sparse image coding. We consider alpha- and beta-divergences as error (cost) functions and derive three different algorithms: (1) multiplicative updating; (2) fixed point alternating least squares (FPALS); (3) alternating interior-point gradient (AIPG) algorithm. We also incorporate these algorithms into multilayer networks. Experimental results confirm the very useful behavior of our multilayer 3D NTF algorithms with multi-start initializations.

## 1 Models and Problem Formulation

Tensors (also known as n-way arrays or multidimensional arrays) are used in a variety of applications ranging from neuroscience and psychometrics to chemometrics [1,2,3,4]. Nonnegative matrix factorization (NMF), Non-negative tensor factorization (NTF), parallel factor analysis PARAFAC and TUCKER models with non-negativity constraints have been recently proposed as promising sparse and quite efficient representations of signals, images, or general data [1,2,3,4,5,6,7,8,9,10,11,12,13,14]. From a viewpoint of data analysis, NTF is very attractive because it takes into account spacial and temporal correlations between variables more accurately than 2D matrix factorizations, such as NMF, and it provides usually sparse common factors or hidden (latent) components

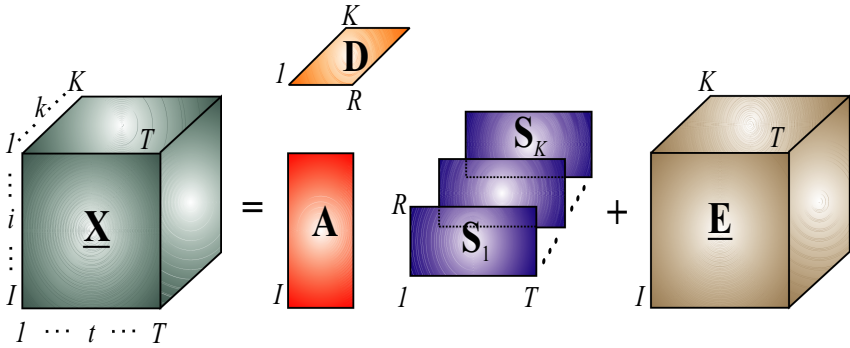
---

\* Invited paper.

\*\* On leave from Warsaw University of Technology, Dept. of EE, Warsaw, Poland.

\*\*\* On leave from Institute of Telecommunications, Teleinformatics and Acoustics, Wroclaw University of Technology, Poland.





**Fig. 1.** NTF model that decomposes approximately tensor  $\underline{\mathbf{X}} \in \mathbb{R}_+^{I \times T \times K}$  to set of nonnegative matrices  $\mathbf{A} = [a_{ir}] \in \mathbb{R}_+^{I \times R}$ ,  $\mathbf{D} \in \mathbb{R}^{K \times R}$  and  $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_K\}$ ,  $\mathbf{S}_k = [s_{rtk}] \in \mathbb{R}_+^{R \times T}$ ,  $\underline{\mathbf{E}} \in \mathbb{R}^{I \times T \times K}$  is a tensor representing errors

with physical or physiological meaning and interpretation [4]. One of our motivations is to develop flexible NTF algorithms which can be applied in neuroscience (analysis of EEG, fMRI) [8,15,16].

The basic 3D NTF model considered in this paper is illustrated in Fig. 1 (see also [9]). A given tensor  $\underline{\mathbf{X}} \in \mathbb{R}_+^{I \times T \times K}$  is decomposed as a set of matrices  $\mathbf{A} \in \mathbb{R}_+^{I \times R}$ ,  $\mathbf{D} \in \mathbb{R}_+^{K \times R}$  and the 3D tensor with the frontal slices  $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_K\}$  with nonnegative entries. Here and elsewhere,  $\mathbb{R}_+$  denotes the nonnegative orthant with appropriate dimensions. The three-way NTF model is given by

$$\mathbf{X}_k = \mathbf{A} \mathbf{D}_k \mathbf{S}_k + \mathbf{E}_k, \quad (k = 1, 2, \dots, K) \tag{1}$$

where  $\mathbf{X}_k = \mathbf{X}_{:, :, k} \in \mathbb{R}_+^{I \times T}$  are the frontal slices of  $\underline{\mathbf{X}} \in \mathbb{R}_+^{I \times T \times K}$ ,  $K$  is a number of vertical slices,  $\mathbf{A} = [a_{ir}] \in \mathbb{R}_+^{I \times R}$  is the basis (mixing matrix) representing common factors,  $\mathbf{D}_k \in \mathbb{R}_+^{R \times R}$  is a diagonal matrix that holds the  $k$ -th row of the matrix  $\mathbf{D} \in \mathbb{R}_+^{K \times R}$  in its main diagonal, and  $\mathbf{S}_k = [s_{rtk}] \in \mathbb{R}_+^{R \times T}$  are matrices representing sources (or hidden components), and  $\mathbf{E}_k = \mathbf{E}_{:, :, k} \in \mathbb{R}_+^{I \times T}$  is the  $k$ -th vertical slice of the tensor  $\underline{\mathbf{E}} \in \mathbb{R}_+^{I \times T \times K}$  representing errors or noise depending upon the application. Typically, for BSS problems  $T \gg I \geq K > R$ . The objective is to estimate the set of matrices  $\mathbf{A}$ ,  $\mathbf{D}$  and  $\{\mathbf{S}_1, \dots, \mathbf{S}_K\}$  subject to some non-negativity constraints and other possible natural constraints such as sparseness and/or smoothness on the basis of only  $\underline{\mathbf{X}}$ . Since the diagonal matrices  $\mathbf{D}_k$  are scaling matrices, they can usually be absorbed by the matrices  $\mathbf{S}_k$  by introducing row-normalized matrices  $\tilde{\mathbf{S}}_k := \mathbf{D}_k \mathbf{S}_k$ , hence  $\mathbf{X}_k = \mathbf{A} \tilde{\mathbf{S}}_k + \mathbf{E}_k$ . Thus in BSS applications the matrix  $\mathbf{A}$  and the set of scaled source matrices  $\tilde{\mathbf{S}}_1, \dots, \tilde{\mathbf{S}}_K$  need only to be estimated. Throughout this paper, we use the following notation: the  $ir$ -th element of the matrix  $\mathbf{A}$  is denoted by  $a_{ir}$ ,  $x_{itk} = [\mathbf{X}_k]_{it}$  means the  $it$ -th element of the  $k$ -th frontal slice  $\mathbf{X}_k$ ,  $s_{rtk} = [\mathbf{S}_k]_{rt}$ ,  $\tilde{\mathbf{S}} = [\tilde{\mathbf{S}}_1, \tilde{\mathbf{S}}_2, \dots, \tilde{\mathbf{S}}_K] \in \mathbb{R}_+^{R \times KT}$  is a row-wise unfolded matrix of the slices  $\tilde{\mathbf{S}}_k$ ,

analogously,  $\bar{\mathbf{X}} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K] \in \mathbb{R}_+^{I \times KT}$  is a row-wise unfolded matrix of the slices  $\mathbf{X}_k$  and  $\bar{x}_{ip} = [\bar{\mathbf{X}}]_{ip}$ ,  $\bar{s}_{rt} = [\bar{\mathbf{S}}]_{rt}$ .

## 2 Cost Functions and Associated NTF Algorithms

To deal with the factorization problem (1) efficiently we adopt several approaches from constrained optimization and multi-criteria optimization, where we minimize simultaneously several cost functions using alternating switching between sets of parameters [5,6,11,12,13]. Alpha and Beta divergences are two complimentary cost functions [6,7,17]. Both divergences build up a wide class of generalized cost functions which can be applied for NMF and NTF [8,7].

### 2.1 NTF Algorithms Using $\alpha$ -Divergence

Let us consider a general class of cost functions, called  $\alpha$ -divergence [6,17]:

$$D^{(\alpha)}(\bar{\mathbf{X}} \parallel \mathbf{A}\bar{\mathbf{S}}) = \frac{1}{\alpha(\alpha - 1)} \sum_{ip} (\bar{x}_{ip} [\mathbf{A}\bar{\mathbf{S}}]_{ip}^{1-\alpha} - \alpha \bar{x}_{ip} + (\alpha - 1) [\mathbf{A}\bar{\mathbf{S}}]_{ip}) \quad (2)$$

$$D_k^{(\alpha)}(\mathbf{X}_k \parallel \mathbf{A}\mathbf{S}_k) = \frac{1}{\alpha(\alpha - 1)} \sum_{itk} (x_{itk} [\mathbf{A}\mathbf{S}_k]_{it}^{1-\alpha} - \alpha x_{itk} + (\alpha - 1) [\mathbf{A}\mathbf{S}_k]_{it}). \quad (3)$$

We note that as special cases of the  $\alpha$ -divergence for  $\alpha = 2, 0.5, -1$ , we obtain the Pearson’s, Hellinger’s and Neyman’s chi-square distances, respectively, while for the cases  $\alpha = 1$  and  $\alpha = 0$  the divergence has to be defined by the limits:  $\alpha \rightarrow 1$  and  $\alpha \rightarrow 0$ , respectively. When these limits are evaluated one obtains for  $\alpha \rightarrow 1$  the generalized Kullback-Leibler divergence (I-divergence) and for  $\alpha \rightarrow 0$  the dual generalized KL divergence [6,7,8].

Instead of applying the standard gradient descent method, we use the nonlinearly transformed gradient descent approach which can be considered as a generalization of the exponentiated gradient (EG):

$$\Phi(s_{rtk}) \leftarrow \Phi(s_{rtk}) - \eta_{rtk} \frac{\partial D_A(\mathbf{X}_k \parallel \mathbf{A}\mathbf{S}_k)}{\partial \Phi(s_{rtk})}, \quad \Phi(a_{ir}) \leftarrow \Phi(a_{ir}) - \eta_{ir} \frac{\partial D_A(\bar{\mathbf{X}} \parallel \mathbf{A}\bar{\mathbf{S}})}{\partial \Phi(a_{ir})},$$

where  $\Phi(x)$  is a suitably chosen function.

It can be shown that such a nonlinear scaling or transformation provides a stable solution and the gradients are much better behaved in the  $\Phi$  space. In our case, we employ  $\Phi(x) = x^\alpha$  (for  $\alpha \neq 0$ ) and choose the learning rates as follows

$$\eta_{rtk} = \alpha^2 \Phi(s_{rtk}) / (s_{rtk}^{1-\alpha} \sum_{i=1}^I a_{ir}), \quad \eta_{ir} = \alpha^2 \Phi(a_{ir}) / (a_{ir}^{1-\alpha} \sum_{p=1}^{KT} \bar{s}_{rp}), \quad (4)$$

which leads directly to the new learning algorithm [1]: (the rigorous proof of local convergence similar to this given by Lee and Seung [13] is omitted due to a lack of space):

<sup>1</sup> For  $\alpha = 0$  instead of  $\Phi(x) = x^\alpha$ , we have used  $\Phi(x) = \ln(x)$ , which leads to a generalized SMART algorithm:  $s_{rtk} \leftarrow s_{rtk} \prod_{i=1}^I (x_{itk} / [\mathbf{A}\mathbf{S}_k]_{it})^{\eta_{ra_{ir}}}$  and  $a_{ir} \leftarrow a_{ir} \prod_{p=1}^{KT} (\bar{x}_{ip} / [\mathbf{A}\bar{\mathbf{S}}]_{ip})^{\eta_{r\bar{s}_{rp}}}$  [7].

$$s_{rtk} \leftarrow s_{rtk} \left( \frac{\sum_{i=1}^I a_{ir} (x_{itk}/[\mathbf{A}\mathbf{S}_k]_{it})^\alpha}{\sum_{q=1}^I a_{qr}} \right)^{1/\alpha}, \tag{5}$$

$$a_{ir} \leftarrow a_{ir} \left( \frac{\sum_{p=1}^{KT} (\bar{x}_{ip}/[\mathbf{A}\bar{\mathbf{S}}]_{ip})^\alpha \bar{s}_{rp}}{\sum_{q=1}^{KT} \bar{s}_{rq}} \right)^{1/\alpha}. \tag{6}$$

The sparsity constraints are achieved via suitable nonlinear transformation in the form  $s_{rtk} \leftarrow (s_{rtk})^{1+\gamma}$  where  $\gamma$  is a small coefficient [6].

### 2.2 NTF Algorithms Using $\beta$ -Divergence

The  $\beta$ -divergence can be considered as a general complimentary cost function to  $\alpha$ -divergence defined above [6,7]. Regularized  $\beta$ -divergences for the NTF problem can be defined as follows:

$$D^{(\beta)}(\bar{\mathbf{X}}\|\mathbf{A}\bar{\mathbf{S}}) = \sum_{ip} \left( \bar{x}_{ip} \frac{\bar{x}_{ip}^\beta - [\mathbf{A}\bar{\mathbf{S}}]_{ip}^\beta}{\beta(\beta+1)} + [\mathbf{A}\bar{\mathbf{S}}]_{ip}^\beta \frac{[\mathbf{A}\bar{\mathbf{S}}]_{ip} - \bar{x}_{ip}}{\beta+1} \right) + \alpha_A \|\mathbf{A}\|_{L1}, \tag{7}$$

$$D_k^{(\beta)}(\mathbf{X}_k\|\mathbf{A}\mathbf{S}_k) = \sum_{it} \left( x_{itk} \frac{x_{itk}^\beta - [\mathbf{A}\mathbf{S}_k]_{it}^\beta}{\beta(\beta+1)} + [\mathbf{A}\mathbf{S}_k]_{it}^\beta \frac{[\mathbf{A}\mathbf{S}_k]_{it} - x_{itk}}{\beta+1} \right) + \alpha_{S_k} \|\mathbf{S}_k\|_{L1}, \tag{8}$$

for  $i = 1, \dots, I$ ,  $t = 1, \dots, T$ ,  $k = 1, \dots, K$ ,  $p = 1, \dots, KT$ , where  $\alpha_{S_k}$  and  $\alpha_A$  are small positive regularization parameters which control the degree of sparseness of the matrices  $\mathbf{S}$  and  $\mathbf{A}$ , respectively, and the  $L1$ -norms defined as  $\|\mathbf{A}\|_{L1} = \sum_{ir} |a_{ir}| = \sum_{ir} a_{ir}$  and  $\|\mathbf{S}_k\|_{L1} = \sum_{rt} |s_{rtk}| = \sum_{rt} s_{rtk}$  are introduced to enforce a sparse representation of the solution. It is interesting to note that in the special case for  $\beta = 1$  and  $\alpha_A = \alpha_{S_k} = 0$ , we obtain the square Euclidean distance expressed by the Frobenius norm  $\|\mathbf{X}_k - \mathbf{A}\mathbf{S}_k\|_F^2$ , while for the singular cases,  $\beta = 0$  and  $\beta = -1$ , the unregularized  $\beta$ -divergence has to be defined as limiting cases as  $\beta \rightarrow 0$  and  $\beta \rightarrow -1$ , respectively. When these limits are evaluated one gets for  $\beta \rightarrow 0$  the generalized Kullback-Leibler divergence (I-divergence) and for  $\beta \rightarrow -1$  we obtain the Itakura-Saito distance.

The choice of the  $\beta$  parameter depends on a statistical distribution of the data and the  $\beta$ -divergence corresponds to the Tweedie models [17]. For example, the optimal choice of the parameter for the normal distribution is  $\beta = 1$ , for the gamma distribution is  $\beta \rightarrow -1$ , for the Poisson distribution  $\beta \rightarrow 0$ , and for the compound Poisson  $\beta \in (-1, 0)$ . By minimizing the above formulated  $\beta$ -divergences, we can derive various kinds of NTF algorithms: Multiplicative based on the standard gradient descent, Exponentiated Gradient (EG), Projected Gradient (PG), Alternating Interior-Point Gradient (AIPG), or Fixed Point (FP) algorithms. By using the standard gradient descent, we obtain the multiplicative update rules:

$$s_{rtk} \leftarrow s_{rtk} \frac{\sum_{i=1}^I a_{ir} (x_{itk}/[\mathbf{A}\mathbf{S}_k]_{it}^{1-\beta}) - \alpha_{S_k}]_\varepsilon}{\sum_{i=1}^I a_{ir} [\mathbf{A}\mathbf{S}_k]_{it}^\beta}, \tag{9}$$

$$a_{ir} \leftarrow a_{ir} \frac{\sum_{p=1}^{KT} (\bar{x}_{ip}/[\mathbf{A}\bar{\mathbf{S}}]_{ip}^{1-\beta}) \bar{s}_{rp} - \alpha_A]_\varepsilon}{\sum_{p=1}^{KT} [\mathbf{A}\bar{\mathbf{S}}]_{ip}^\beta \bar{s}_{rp}}, \tag{10}$$

where the half-wave rectification defined as  $[x]_\varepsilon = \max\{\varepsilon, x\}$  with a positive small  $\varepsilon = 10^{-16}$  is introduced in order to avoid zero and negative values.

In the special case, for  $\beta = 1$ , we can derive a new alternative algorithm referred to as, FPALS (Fixed Point Alternating Least Squares) algorithm [8]:

$$\mathbf{S}_k \leftarrow \left[ (\mathbf{A}^T \mathbf{A} + \gamma_A \mathbf{E})^+ (\mathbf{A}^T \mathbf{X}_k - \alpha_{S_k} \mathbf{E}_S) \right]_\varepsilon, \tag{11}$$

$$\mathbf{A} \leftarrow \left[ (\bar{\mathbf{X}} \bar{\mathbf{S}}^T - \alpha_A \mathbf{E}_A) (\bar{\mathbf{S}} \bar{\mathbf{S}}^T + \gamma_S \mathbf{E})^+ \right]_\varepsilon, \tag{12}$$

where  $\mathbf{A}^+$  denotes Moore-Penrose pseudo-inverse,  $\mathbf{E} \in \mathbb{R}^{R \times R}$ ,  $\mathbf{E}_S \in \mathbb{R}^{R \times T}$  and  $\mathbf{E}_A \in \mathbb{R}^{I \times R}$  are matrices with all ones and the function  $[\mathbf{X}]_\varepsilon = \max\{\varepsilon, \mathbf{X}\}$  is componentwise. The above algorithm can be considered as a nonlinear projected Alternating Least Squares (ALS) or nonlinear extension of the EM-PCA algorithm[2].

Furthermore, using the Alternating Interior-Point Gradient (AIPG) approach [18], another new efficient algorithm has been derived:

$$\mathbf{S}_k \leftarrow \mathbf{S}_k - \eta_{S_k} \mathbf{P}_{S_k}, \quad \mathbf{P}_{S_k} = \left( \mathbf{S}_k \oslash (\mathbf{A}^T \mathbf{A} \mathbf{S}_k) \right) \odot \left( \mathbf{A}^T (\mathbf{A} \mathbf{S}_k - \mathbf{X}_k) \right), \tag{13}$$

$$\mathbf{A} \leftarrow \mathbf{A} - \eta_A \mathbf{P}_A, \quad \mathbf{P}_A = \left( \mathbf{A} \oslash (\mathbf{A} \bar{\mathbf{S}} \bar{\mathbf{S}}^T) \right) \odot \left( (\mathbf{A} \bar{\mathbf{S}} - \bar{\mathbf{X}}) \bar{\mathbf{S}}^T \right), \tag{14}$$

where the operators  $\odot$  and  $\oslash$  mean component-wise multiplication and division, respectively. The learning rates  $\eta_{S_k}$  and  $\eta_A$  are selected in this way to ensure the steepest descent, and on the other hand, to maintain non-negativity. Thus,  $\eta_{S_k} = \min\{\tau \hat{\eta}_{S_k}, \eta_{S_k}^*\}$  and  $\eta_A = \min\{\tau \hat{\eta}_A, \eta_A^*\}$ , where  $\tau \in (0, 1)$ ,  $\hat{\eta}_{S_k} = \{\eta : \mathbf{S}_k - \eta \mathbf{P}_{S_k}\}$  and  $\hat{\eta}_A = \{\eta : \mathbf{A} - \eta \mathbf{P}_A\}$  ensure non-negativity, and

$$\eta_{S_k}^* = \frac{\text{vec}(\mathbf{P}_{S_k})^T \text{vec}(\mathbf{A}^T \mathbf{A} \mathbf{S}_k - \mathbf{A}^T \mathbf{X}_k)}{\text{vec}(\mathbf{A} \mathbf{P}_{S_k})^T \text{vec}(\mathbf{A} \mathbf{P}_{S_k})}, \quad \eta_A^* = \frac{\text{vec}(\mathbf{P}_A)^T \text{vec}(\mathbf{A} \bar{\mathbf{S}} \bar{\mathbf{S}}^T - \bar{\mathbf{X}} \bar{\mathbf{S}}^T)}{\text{vec}(\mathbf{P}_A \bar{\mathbf{S}})^T \text{vec}(\mathbf{P}_A \bar{\mathbf{S}})}$$

are the adaptive steepest descent learning rates [8].

### 3 Multi-layer NTF

In order to improve the performance of all the developed NTF algorithms, especially for ill-conditioned and badly scaled data and also to reduce risk of getting

<sup>2</sup> In order to drive the modified FPALS algorithm, we have used the following regularized cost functions:  $\|\mathbf{X}_k - \mathbf{A}\mathbf{S}_k\|_F^2 + \alpha_{S_k} \|\mathbf{S}_k\|_{L_1} + \gamma_S \text{tr}\{\mathbf{S}_k^T \mathbf{E} \mathbf{S}_k\}$  and  $\|\bar{\mathbf{X}} - \mathbf{A}\bar{\mathbf{S}}\|_F^2 + \alpha_A \|\mathbf{A}\|_{L_1} + \gamma_A \text{tr}\{\mathbf{A} \mathbf{E} \mathbf{A}^T\}$ , where  $\gamma_S, \gamma_A$  are nonnegative regularization coefficients imposing some kinds of smoothness and sparsity.

stuck in local minima in non-convex alternating minimization computations, we have developed a simple hierarchical and multi-stage procedure combined together with multi-start initializations, in which we perform a sequential decomposition of nonnegative matrices as follows. In the first step, we perform the basic decomposition (factorization)  $\mathbf{X}_k = \mathbf{A}^{(1)} \mathbf{S}_k^{(1)}$  using any available NTF algorithm. In the second stage, the results obtained from the first stage are used to perform the similar decomposition:  $\mathbf{S}_k^{(1)} = \mathbf{A}^{(2)} \mathbf{S}_k^{(2)}$  using the same or different update rules, and so on. We continue our decomposition taking into account only the last achieved components. The process can be repeated arbitrarily many times until some stopping criteria are satisfied. In each step, we usually obtain gradual improvements of the performance. Thus, our NTF model has the form:  $\mathbf{X}_k = \mathbf{A}^{(1)} \mathbf{A}^{(2)} \dots \mathbf{A}^{(L)} \mathbf{S}_k^{(L)}$ , with the basis nonnegative matrix defined as  $\mathbf{A} = \mathbf{A}^{(1)} \mathbf{A}^{(2)} \dots \mathbf{A}^{(L)}$ . Physically, this means that we build up a system that has many layers or cascade connections of  $L$  mixing sub-systems. The key point in our novel approach is that the learning (update) process to find parameters of sub-matrices  $\mathbf{S}_k^{(l)}$  and  $\mathbf{A}^{(l)}$  is performed sequentially, i.e. layer by layer. This can be expressed by the following procedure [78,19]:

**Outline Multilayer NTF Algorithm**

Initialize randomly  $\mathbf{A}^{(l)}$  and/or  $\mathbf{S}_k^{(l)}$  and perform the alternating minimization till convergence:

$$\mathbf{S}_k^{(l)} \leftarrow \arg \min_{\mathbf{S}_k^{(l)} \geq 0} \left\{ D_k \left( \mathbf{S}_k^{(l-1)} \parallel \mathbf{A}^{(l)} \mathbf{S}_k^{(l)} \right) \right\}, \quad k = 1, \dots, K, \quad \bar{\mathbf{S}}^{(l)} = [\mathbf{S}_1^{(l)}, \dots, \mathbf{S}_K^{(l)}],$$

$$\mathbf{A}^{(l)} \leftarrow \arg \min_{\mathbf{A}^{(l)} \geq 0} \left\{ \tilde{D} \left( \bar{\mathbf{S}}^{(l-1)} \parallel \mathbf{A}^{(l)} \bar{\mathbf{S}}^{(l)} \right) \right\}, \quad [\mathbf{A}^{(l)}]_{ir} \leftarrow \left[ a_{ir} / \sum_{i=1}^I a_{ir} \right]^{(l)},$$

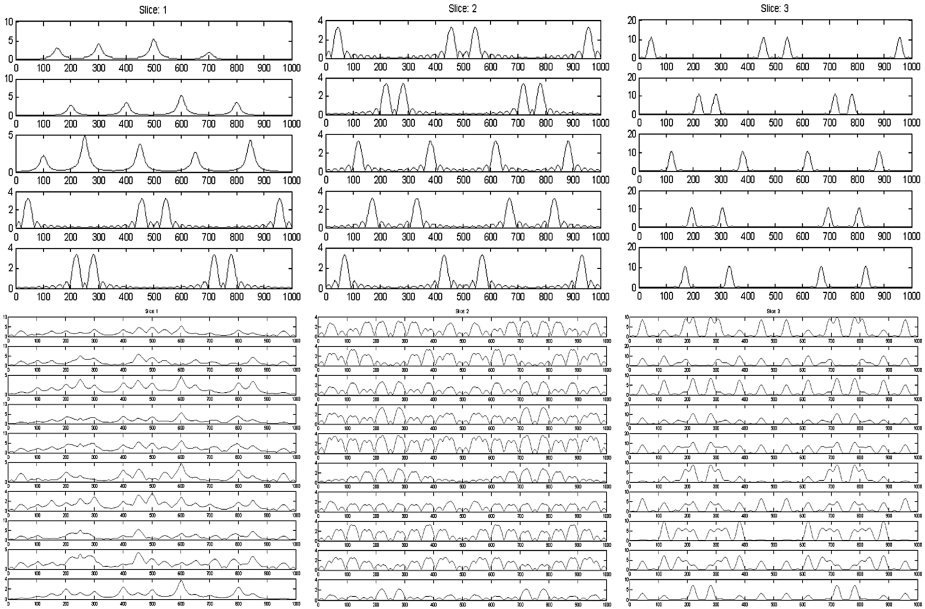
$$\mathbf{S}_k = \mathbf{S}_k^{(L)}, \quad \mathbf{A} = \mathbf{A}^{(1)} \dots \mathbf{A}^{(L)},$$

where  $D_k$  and  $\tilde{D}$  are the cost functions (not necessary identical) used for estimation of  $\mathbf{S}_k$  and  $\mathbf{A}$ , respectively.

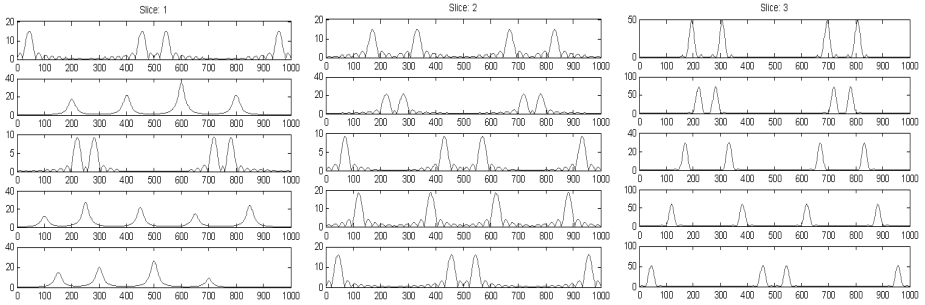
An open theoretical issue is to prove mathematically or explain more rigorously why the multilayer distributed NTF system with multi-start initializations results in considerable improvement in performance and reduces the risk of getting stuck in local minima. An intuitive explanation is as follows: the multilayer system provides a sparse distributed representation of basis matrices  $\mathbf{A}^{(l)}$ , so even a true basis matrix  $\mathbf{A}$  is not sparse it can be represented by a product of sparse factors. In each layer we force (or encourage) a sparse representation. We found by extensive experiments that if the true basis matrix is sparse, most standard NTF/NMF algorithms have improved performance (see next section). However, in practice not all data provides a sufficiently sparse representation, so the main idea is to model any data by cascade connections of sparse sub-systems. On the other hand, such multilayer systems are biologically motivated and plausible.

### 4 Simulation Results

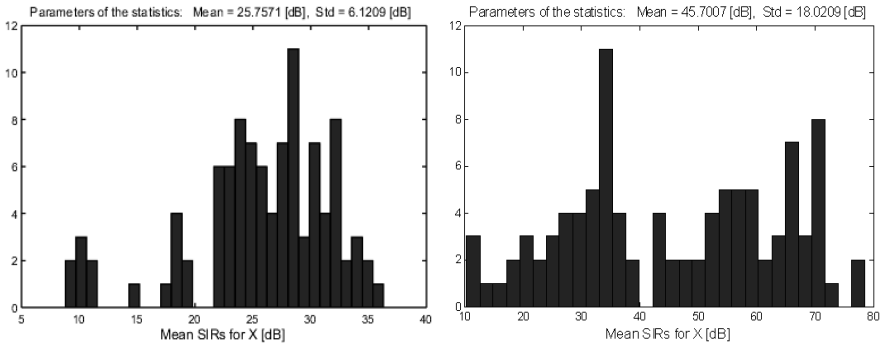
All the NMF algorithms presented in this paper have been extensively tested for many difficult benchmarks for signals and images with various statistical distributions of signals and additive noise, and also for preliminary tests with real EEG data. Due to space limitations we present here only comparison of proposed algorithms for a typical benchmark. The simulations results shown in Table 1 have been performed for the synthetic benchmark in which the nonnegative weakly statistically dependent 100 hidden components or sources (spectra) are collected in 20 slices  $\mathbf{S}_k \in \mathbb{R}_+^{5 \times 1000}$ , each representing 5 different kind of spectra. The sources have been mixed by the common random matrix  $\mathbf{A} \in \mathbb{R}_+^{10 \times 5}$  with a uniform distribution. In this way, we obtained the 3D tensor  $\mathbf{X} \in \mathbb{R}^{10 \times 1000 \times 20}$  of overlapped spectra. Table 1 shows the averaged SIR (standard signal to interference ratio) performance obtained from averaging the results from 100 runs of the Monte Carlo (MC) analysis for recovering of the original spectra  $\mathbf{S}_k$  and the mixing matrix  $\mathbf{A}$  for various algorithms and for different number of layers 1-5. (Usually, it assumed that  $SIR \geq 20dB$  provides a quite good performance, and over  $30dB$  excellent performance.) We have also applied and tested the developed algorithms for real-world EEG data and neuroimages. Due to space limitation these results will be presented in the conference and on our website.



**Fig. 2.** Selected slices of: (top) the original spectra signals (top); mixed signals with dense mixing matrix  $\mathbf{A} \in \mathbb{R}^{10 \times 5}$



**Fig. 3.** Spectra signals estimated with the FPALS (11)–(12) using 3 layers for  $\gamma_A = \gamma_S = \alpha_{S_k} = \alpha_A = 0$ . The signals in the corresponding slices are scored with: SIRs(1-st slice) = 47.8, 53.6, 22.7, 43.3, 62; SIRs(2-nd slice) = 50, 52.7, 23.6, 42.5, 62.7; and SIRs(3-d slice) = 50.1, 55.8, 30, 46.3, 59.9; [dB], respectively.



**Fig. 4.** Histograms of 100 mean-SIR samples from Monte Carlo analysis performed using the following algorithms with 5 layers: (left) Beta Alg. (9)–(10),  $\beta = 0$ ; (right) FPALS (11)–(12) for  $\gamma_A = \gamma_S = \alpha_{S_k} = \alpha_A = 0$ .

**Table 1.** Mean SIRs in [dB] obtained from 100 MC samples for estimation of the columns in  $\mathbf{A}$  and the rows (sources) in  $\mathbf{S}_k$  versus the number of layers (Multi-layer technique), and for the selected algorithms

ALGORITHMS: (Equations)	LAYERS (SIRs $\mathbf{A}$ )					LAYERS (SIRs $\mathbf{S}$ )				
	1	2	3	4	5	1	2	3	4	5
Alpha Alg. (5–6): $\alpha = 0.5$	9.1	15.6	19	21.8	24.6	7.8	13.5	16.5	18.9	21.2
Beta Alg. (9–10): $\beta = 0$	11.9	20.9	27.8	29.5	30.8	8.1	16.4	22.9	24.4	25.6
AIPG (13–14)	14	22.7	29	33.1	35.4	10.1	18	24.1	28.4	30.6
FPALS (11–12)	20.7	35	42.6	46	47.2	19.4	32.7	41.7	46.1	48.1

**Table 2.** Elapsed times (in seconds) for 1000 iterations with different algorithms

No. layers	Alpha Alg. (5–6) $\alpha = 0.5$	Beta Alg. (9–10) $\beta = 0$	AIPG (13–14)	FPALS (11–12)
1	23.7	4.7	11.8	3.8
3	49.3	11.3	32.8	10.3

## 5 Conclusions and Discussion

The main objective and motivations of this paper was to develop and compare leaning algorithms and compare their performance. We have extended the 3D non-negative matrix factorization (NMF) models to multi-layer models and found that the best performance is obtained with the FPALS and AIPG algorithms. With respect to the standard NTF (single layer) models, our model and proposed algorithms can give much better performance or precision in factorizations and better robustness against noise. Moreover, we considered a wide class of the cost functions which allows us to derive a family of robust and efficient NTF algorithms with only single parameter to tune ( $\alpha$  or  $\beta$ ). The optimal choice of the parameter in the cost function depends and on a statistical distribution of data and additive noise, thus different criteria and algorithms (updating rules) should be applied for estimating the basis matrix  $\mathbf{A}$  and the source matrices  $\mathbf{S}_k$ , depending on *a priori* knowledge about the statistics of noise or errors. We found by extensive simulations that the multi-layer technique combined together with multi-start initialization plays a key role in improving the performance of blind source separation when using the NTF approach. It is worth mentioning that we can use two different strategies. In the first approach presented in details in this contribution we use two different cost functions: A global cost function (using row-wise unfolded matrices:  $\bar{\mathbf{X}}$ ,  $\bar{\mathbf{S}}$  and 2D model  $\bar{\mathbf{X}} = \mathbf{A}\bar{\mathbf{S}}$ ) to estimate the common factors, i.e., the basis (mixing) matrix  $\mathbf{A}$ ; and local cost functions to estimate the frontal slices  $\mathbf{S}_k$ , ( $k = 1, 2, \dots, K$ ). However, it is possible to use a different approach in which we use only a set of local cost functions, e.g.,  $D_k = 0.5\|\mathbf{X}_k - \mathbf{A}\mathbf{S}_k\|_F^2$ . In such a case, we estimate  $\mathbf{A}$  and  $\mathbf{S}_k$  cyclically by applying alternating minimization (similar to row-action projection in the Kaczmarz algorithm). We found that such approach also works well for the NTF model. We have motivated the use of proposed 3D NTF in three areas of data analysis (especially, EEG and fMRI) and signal/image processing: (i) multi-way blind source separation, (ii) model reductions and selection, and (iii) sparse image coding. Our preliminary experiments are promising.

The proposed models can be further extended by imposing additional, natural constraints such as smoothness, continuity, closure, unimodality, local rank - selectivity, and/or by taking into account a prior knowledge about specific 3D, or more generally, multi-way data.

Obviously, there are many challenging open issues remaining, such as global convergence, an optimal choice of the parameters and the model.



## References

1. Workshop on tensor decompositions and applications, CIRM, Marseille, France (2005)
2. Hazan, T., Polak, S., Shashua, A.: Sparse image coding using a 3D non-negative tensor factorization. In: International Conference of Computer Vision (ICCV). (2005) 50–57
3. Heiler, M., Schnoerr, C.: Controlling sparseness in nonnegative tensor factorization. In: ECCV. (2006)
4. Smilde, A., Bro, R., Geladi, P.: Multi-way Analysis: Applications in the Chemical Sciences. John Wiley and Sons, New York (2004)
5. Berry, M., Browne, M., Langville, A., Pauca, P., Plemmons, R.: Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics and Data Analysis (**2006**) submitted.
6. Cichocki, A., Zdunek, R., Amari, S.: Csiszar’s divergences for non-negative matrix factorization: Family of new algorithms. LNCS **3889** (2006) 32–39
7. Cichocki, A., Amari, S., Zdunek, R., Kompass, R., Hori, G., He, Z.: Extended SMART algorithms for non-negative matrix factorization. LNAI **4029** (2006) 548–562
8. Cichocki, A., Zdunek, R.: NMFLAB for Signal and Image Processing. Technical report, Laboratory for Advanced Brain Signal Processing, BSI, RIKEN, Saitama, Japan (2006)
9. Cichocki, A., Zdunek, R.: NTFLAB for Signal Processing. Technical report, Laboratory for Advanced Brain Signal Processing, BSI, RIKEN, Saitama, Japan (2006)
10. Dhillon, I., Sra, S.: Generalized nonnegative matrix approximations with Bregman divergences. In: Neural Information Proc. Systems, Vancouver, Canada (2005)
11. Hoyer, P.: Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research **5** (2004) 1457–1469
12. Kim, M., Choi, S.: Monaural music source separation: Nonnegativity, sparseness, and shift-invariance. LNCS **3889** (2006) 617–624
13. Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. Nature **401** (1999) 788–791
14. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: Tensorfaces, Copenhagen, Denmark, Proc. European Conf. on Computer Vision (ECCV) (2002) 447–460
15. Morup, M., Hansen, L.K., Herrmann, C.S., Parnas, J., Arnfred, S.M.: Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG. NeuroImage **29** (2006) 938–947
16. Miwakeichi, F., Martinez-Montes, E., Valds-Sosa, P.A., Nishiyama, N., Mizuhara, H., Yamaguchi, Y.: Decomposing EEG data into spacetime-frequency components using Parallel Factor Analysis. NeuroImage **22** (2004) 1035–1045
17. Amari, S.: Differential-Geometrical Methods in Statistics. Springer Verlag (1985)
18. Merritt, M., Zhang, Y.: An interior-point gradient method for large-scale totally nonnegative least squares problems. J. Optimization Theory and Applications **126** (2005) 191–202
19. Cichocki, A., Zdunek, R.: Multilayer nonnegative matrix factorization. Electronics Letters **42** (2006) 947–948

# A Real-Time Adaptive Wavelet Transform-Based QRS Complex Detector

Marek Rudnicki and Paweł Strumiłło

Institute of Electronics, Technical University of Łódź,  
211/215 Wólczańska, 90-924 Łódź, Poland  
marekrud@gmail.com, pawel.strumillo@p.lodz.pl

**Abstract.** In this paper, the design and test results of a QRS complex detector are presented. The detection algorithm is based on the Discrete Wavelet Transform and implements an adaptive weighting scheme of the selected transform coefficients in the time domain. It was tested against a standard MIT-BIH Arrhythmia Database of ECG signals for which sensitivity ( $Se$ ) of 99.54% and positive predictivity ( $+P$ ) of 99.52% was achieved. The designed QRS complex detector is implemented on TI TMS320C6713 DSP for real-time processing of ECGs.

## 1 Introduction

The electrocardiogram (ECG) is a diagnostic signal recorded at standard electrode locations on patients body that reflects electrical activity of cardiac muscles. The dominant feature of the ECG is the pulse like waveform — termed the QRS complex — that corresponds to the vital time instance at which the heart ventricles are depolarised. The QRS complex (typically lasting no longer than 0.1 s) serves as a reference point for most ECG signal processing algorithms, e.g. it is used in heart rate variability analysers, arrhythmia monitors, implantable pacemakers and ECG signal compression techniques. Hence, the more reliable and precise QRS detection, the better quality of ECG analysers can be expected. However, due to inherent ECG signal variability and different sources corrupting it (e.g. power line and RF interferences, muscle artifacts), QRS complex detection is not a trivial task and still remains an important research topic in the computerised ECG analysis [1].

Most of the QRS complex detection algorithms share similar structure that comprises the two major computing stages: *signal pre-processing* and *data classification*. The role of the pre-processing stage is to extract signal features that are most relevant to the QRS complex. Then these features are fed to the classification module where the decision about the QRS complex event is undertaken. Approaches within the pre-processing stage are based on linear or nonlinear signal processing techniques (e.g. band-pass filtering, signal transformations, mathematical morphology) whereas in the decision stage, simple threshold-like techniques, statistical methods, or more advanced neuro-fuzzy classifiers are used. See [1] for a comprehensive review and comparison of different algorithms for detecting the QRS complex. The choice of a particular computing technique

for QRS detection depends on whether an on-line or off-line ECG analysis is foreseen. Detection performance of the latter one can be additionally improved by applying the so called back-search techniques.

The main difficulty in constructing robust automated QRS complex detectors is due to non-stationarity of both the signal morphology and noise characteristics. This has precluded successful use of early QRS complex detectors that were based solely on the matched filtering concept. Although, Thakor et al. [2] have identified that main spectral components representing the QRS complex are centred around 17 Hz, approaches based on band-pass filters perform insufficiently due to varying temporal spectral features of ECG waveforms. In recent years it has been shown that the wavelet transform [3] is a suitable representation of ECG signals for identifying QRS complex features. It offers an over-determined multiresolution signal representation in the time-scale domain [4,5,6,7].

The idea behind the method proposed here is to apply a scheme for adaptive tracking of wavelet domain features representing the QRS complex so that its detection performance can be continuously maximised. The method was tested both in off-line and on-line DSP implementations [8] on the MIT-BIH Arrhythmia Database annotated recordings [9].

## 2 Description of the Algorithm

The structure of the algorithm is shown in Fig. 1. It consists of the three main processing steps: *pre-processing stage*, *decision stage* and *adaptive weighting of the DWT coefficients*.

Once the ECG signal is transformed into its time-scale representation, the QRS complex detection takes place in this domain. The algorithm is based on adaptive scheme that makes the detection task more robust to continuously varying QRS complex morphology as well as changes in the noise’s bandwidth characteristic.

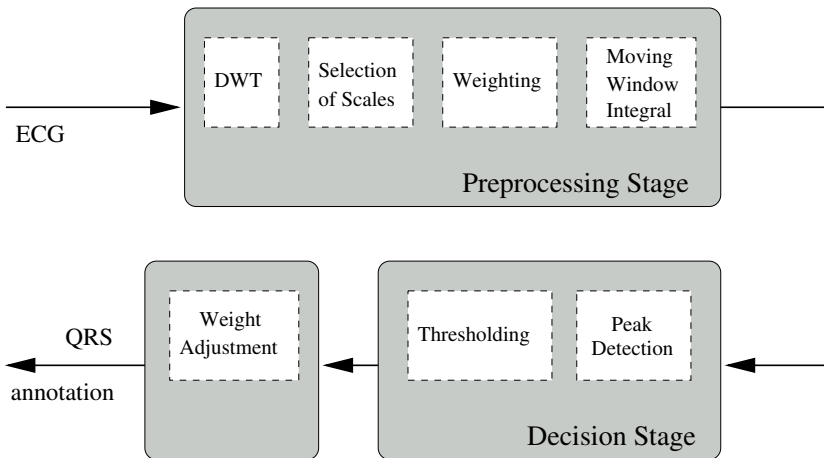


Fig. 1. Block diagram of the proposed QRS complex detector

The developed QRS complex detection algorithm is optimised for MIT-BIH Arrhythmia Database signals, which are sampled at the frequency of 360 Hz.

### 2.1 The Discrete Wavelet Transform

The advantage of the used DWT is that it is possible to obtain good separation of the QRS complexes from other ECG components and noise in the time-scale plane.

The key decision when applying the DWT for signal analysis is the selection of the appropriate prototype wavelet [10]. Because there is no absolute way of selecting the best mother-wavelet function, the tree different wavelets were tested: Haar, Daubechies 4 (D4) and Daubechies 6 (D6) (Fig. 2). The Daubechies wavelet family was considered, because of its compact support and shape resemblance to the shape of QRS complexes [11].

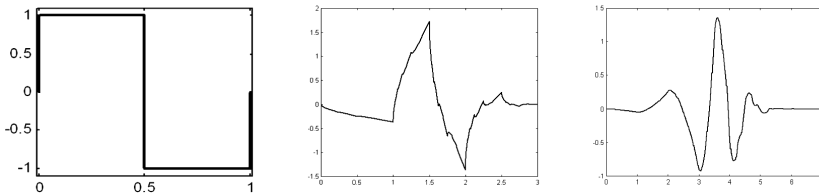


Fig. 2. Haar, Daubechies 4 (D4) and Daubechies 6 (D6) wavelets

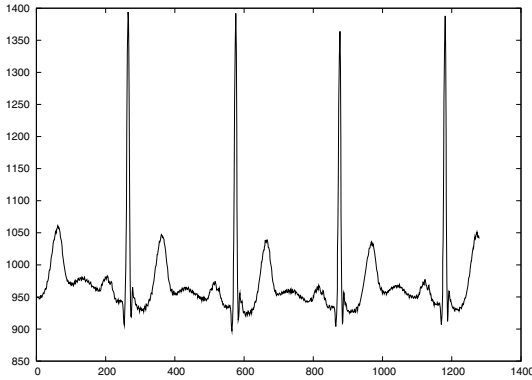
Fig. 3 shows sample ECG signal used to illustrate the QRS complex detection procedure. Its discrete time-scale representation is presented in Fig. 4. Note that the absolute values are displayed.

### 2.2 Selection of Scales and Weighting of the DWT Coefficients

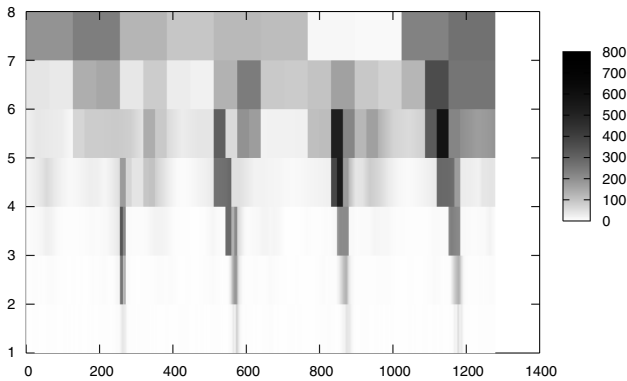
As indicated in Fig. 4 the QRS complex energy is concentrated at some particular scales. Obviously, the DWT decomposition does not provide complete separation of morphological signal features across scales. The scales that mainly reflect the QRS components also capture (to a lesser extent) either noise or signal features such as the P and T waves. By proper selection of those scales for further processing, the ratio between QRS complex energy and energy of other ECG components can be significantly improved. In the described algorithm the DWT coefficients from scales  $2^3$ ,  $2^4$  and  $2^5$  (i.e. 3<sup>rd</sup>, 4<sup>th</sup> and 5<sup>th</sup> decomposition level) are used for the purpose of QRS complex detection. The remaining coefficients are neglected.

In order to emphasise scales that contain more QRS complex energy than the other signal components, weighting of the DWT coefficients is proposed. There is a different weighting coefficient assigned to each scale and its value is calculated according to the following equation:

$$w_i = \frac{S_i^2}{N_i^2} \tag{1}$$



**Fig. 3.** Sample ECG signal (recording no. 103 from MIT-BIH Arrhythmia Database)



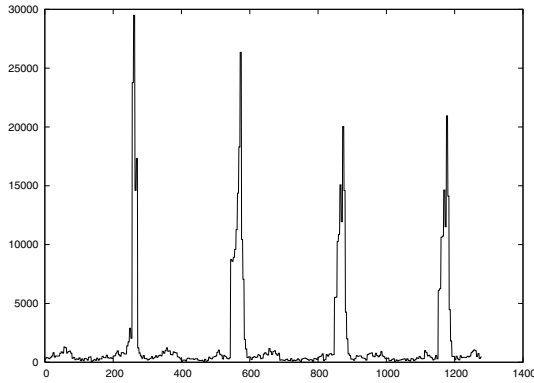
**Fig. 4.** Absolute values of the DWT coefficients obtained for the D4 wavelet for the ECG signal from Fig. 3; the darker the region, the higher the value of the wavelet coefficient

where  $S_i$  is the mean of the DWT coefficients from the  $i$ -th decomposition level within QRS complex interval and  $N_i$  is the mean of the DWT coefficients from the  $i$ -th decomposition level outside of the QRS complex. One can interpret  $w_i$  as a signal-to-noise ratio of the QRS complex to other signal components at a given decomposition level. Therefore,  $S_i^2$  can be seen as the energy of the QRS complex and  $N_i^2$  is equal to the energy of noise and other ECG features.

Next, the DWT coefficients are multiplied by those weighting values and the results are summed up across scales. The following equation describes this procedure:

$$b(n) = \sum_{i=3}^5 w_i \cdot d_i(n) \tag{2}$$

where  $n$  indicates the  $n$ -th signal sample,  $d_i$  are the DWT coefficients at the  $i$ -th decomposition level and  $w_i$  are the corresponding weights. The obtained samples  $b(n)$  are passed to the next processing step of the algorithm. Fig. 5 shows a plot of the processed signal after this intermediate step.



**Fig. 5.** The processed signal after weighting the DWT coefficients and summing them up across the selected scales

The following *moving window filtering* is then used to smooth out multiple peaks corresponding to the QRS complex intervals:

$$y(n) = \frac{1}{N} [b(n - (N - 1)) + b(n - (N - 2)) + \dots + b(n)] \tag{3}$$

where  $N$  is the window width. This parameter should be chosen carefully for good performance of the QRS complex detection [12]. In our algorithm filter order  $N=32$  is used.

### 2.3 Peak Detection and Thresholding

The principle of peak detection is that the monotonicity of the function within a predefined time interval is determined. Whenever it changes from an increasing value to the decreasing one, a new peak is detected and then categorised. The method for peaks classification is based on the algorithm described in [12] with some small modifications. In short, there is a threshold value  $T$  which is compared with values of the detected peaks. If a peak is higher than  $T$ , it is classified as a true QRS peak and as a noise otherwise. Value of  $T$  is recalculated every time a new peak is classified using the following equations:

$$T = P_N + \eta \cdot (P_S - P_N) \tag{4}$$

$$P_S = 0.2 \cdot P + 0.8 \cdot P'_S \quad \text{if } P \text{ is the signal peak} \tag{5}$$

$$P_N = 0.2 \cdot P + 0.8 \cdot P'_N \quad \text{if } P \text{ is the noise peak} \tag{6}$$

where  $P$  is the value of the processed peak;  $P'_S$  is the old value of the running estimate of the signal peak;  $P_S$  is the new value of the running estimate of the signal peak;  $P'_N$  is the old value of the running estimate of the noise peak;  $P_N$  is the new value of the running estimate of the noise peak; and  $\eta$  is the adjustable coefficient. By changing  $\eta$  one can find the optimal ratio between false positive and false negative QRS complex detections. The optimal value of  $\eta = 0.3$  was empirically found for the MIT-BIH recordings and the D4 wavelet was used in the DWT.

Additionally, the so-called *refractory period* is taken into account. During this period that occurs immediately after any QRS complex, the cardiac tissue is not able to respond to any excitation and cause another QRS complex. Length of the refractory period is approx. 0.2 s and all the peaks within this time interval are automatically classified as noise peaks.

### 3 Adaptive Update of Weights

In order to improve the performance of the algorithm, a concept of continuous on-line adjustment of the weights defined in Sect. 2.2 is introduced. Any shape variations of QRS complexes or changes in noise characteristic are reflected in changes of the DWT coefficients. The weights follow those changes and emphasise scales most relevant to the QRS complexes. For example, if any scale is contaminated by noise, the value of the corresponding weight decreases.

Each time a new QRS complex is detected, new values of the weights at the  $i$ -th DWT scale are calculated according to the following equation:

$$w_i = 0.97 \cdot w'_i + 0.03 \cdot v_i \tag{7}$$

where  $w_i$  is the new value of the running estimate of the weight;  $w'_i$  is the old value of the running estimate of the weight and  $v_i$  is the current value of the weight calculated from the most recent DWT coefficients. Flow diagram illustrating this weight update scheme inserted in a feedback loop of the proposed QRS detection algorithm is shown in Fig. 6.

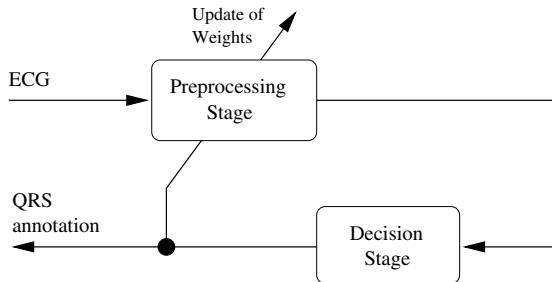


Fig. 6. Flow diagram of the QRS complex detection algorithm

## 4 Results

The algorithm was tested against a standard ECG database, i.e. the MIT-BIH Arrhythmia Database. The database consists of 48 half-hour ECG recordings and contains approximately 109,000 manually annotated signal labels. ECG recordings are two channel, however for the purpose of QRS complex detection only the first channel was used (usually the MLI<sub>I</sub> lead). Database signals are sampled at the frequency of 360 Hz with 11-bit resolution spanning signal voltages within  $\pm 5$  mV range.

QRS complex detection statistic measures were computed by the use of the software from the *Physionet Toolkit* provided with the database. The two most essential parameters we used for describing the overall performance of the QRS complex detector are: *sensitivity*  $Se$  and *positive predictivity*  $+P$ .

Obviously, the QRS complex performance depends on the wavelet function used for the DWT. The best results as shown in Table 1 were obtained for the D4 wavelet. TP stands for the number of true positive detections; FP is the number of false positive detections and FN is the number of false negative detections.

**Table 1.** The overall performance of the proposed QRS complex detection algorithm on MIT-BIH Arrhythmia Database recordings for different types of wavelets

Wavelet used	TP	FP	FN	Se (%)	+P (%)
Haar	90026	1221	1259	98.62	98.66
D4	90864	440	421	99.54	99.52
D6	90793	506	492	99.46	99.45

The algorithm was tested in two additional configurations. In both cases the D4 wavelet was used as the basis function for the DWT. For those tests the adaptive mechanism of weighting coefficients adjustment was turned off. In the first case the weights were set to unity. Whereas in the second case they were set to values obtained after processing of the first 11 minutes of recording no. 119. The normalised values of those coefficients are given below:

$$w_3 = 10.9 \quad w_4 = 8.1 \quad w_5 = 1.0 \tag{8}$$

Recording no. 119 was selected as the regular ECG signal with a very low noise level. Table 2 shows the results.

**Table 2.** The performance of the algorithm without the adaptive adjustment of the weighting coefficients

Weighting coef.	TP	FP	FN	Se (%)	+P (%)
All equal to 1	89281	1770	2004	97.80	98.06
As in (8)	90858	449	427	99.53	99.51



Additionally, the time accuracy of the detections was estimated. The root mean square of the R-R interval error between the reference annotations and the tested annotations is around 20 ms.

## 5 Conclusions

This study has confirmed that the DWT can be successfully applied for the QRS complex detection. For our detector tested on the MIT-BIH Arrhythmia Database ECG recordings, the best results were obtained for the Daubechies 4 wavelet. However, as reported in literature, very good detection results can be also obtained for other wavelet families.

The proposed scheme for weighting the DWT coefficients, as well as adaptive adjustment of these weights, have significantly improved detector performance. By these means the algorithm becomes more robust to variations in the QRS complex morphology and better adapts to changes in the noise characteristics. The achieved detection sensitivity ( $Se$ ) and positive predictivity ( $+P$ ) are 99.54% and 99.52% correspondingly. However, it must be clear that after the weighting coefficients are adjusted to a certain level, the adaptation scheme gives no significant improvements in the performance and the coefficients do not vary substantially.

Detector timing accuracy depends predominantly on the variations in the QRS complex morphology. The R-R interval error between the reference and the obtained annotations can be improved by modifications in the *peak detection* processing step.

Hardware implementation of the algorithm on the TI TMS320C6713 DSP shows that the presented approach can be successfully used in real-time QRS complex detectors [\[8\]](#).

## References

1. Köhler, B.U., Hennig, C., Orhlmeister, R.: The principles of software QRS detection. *Engineering in Medicine and Biology Magazine, IEEE* **21**(1) (2002) 42–57
2. Thakor, N.V., Webster, J.G., Tompkins, W.J.: Estimation of QRS complex power spectra for design of a QRS filter. *IEEE Transactions on Biomedical Engineering* **31**(11) (1984) 702–706
3. Mallat, S.: *A wavelet tour of signal processing*. Academic Press (1998)
4. Kadambe, S., Murray, R., Boudreaux-Bartels, G.F.: Wavelet transform-based QRS complex detector. *IEEE Transactions on Biomedical Engineering* **46**(7) (1999) 838–848
5. Strumillo, P.: Haar wavelet filter for precise detection of the QRS complex in ECGs. In: *5th International Conference on Computers in Medicine Lodz, Poland*. (1999) 150–156
6. Alfonso, V.X., Tompkins, W.J., Nguyen, T.Q., Luo, S.: ECG beat detection using filter banks. *IEEE Transactions on Biomedical Engineering* **46**(2) (1999) 192–202
7. Addison, P.S., Watson, J.N., Clegg, G.R., Holzer, M., Sterz, F., Robertson, C.E.: Evaluating arrhythmias in ECG signals using wavelet transforms. *Engineering in Medicine and Biology Magazine, IEEE* **19**(5) (2000) 104–109

8. Rudnicki, M.: A real-time DSP implementation of wavelet transform-based QRS complex detector. Master's thesis, Technical University of Łódź (2006)
9. Mark, R., Moody, G.: MIT-BIH Arrhythmia Database Directory. MIT (1988) <http://physionet.org/physiobank/>
10. Senhadji, L., Carroult, G., Bellanger, J.J., Passariello, G.: Comparing wavelet transforms for recognizing cardiac patterns. *Engineering in Medicine and Biology Magazine, IEEE* **14**(2) (1995) 167–195
11. Mahmoodabadi, S.Z., Ahmadian, A., Abolhasani, M.D.: ECG feature extraction using Daubechies wavelets. In: *Proceedings of the Fifth IASTED International Conference Visualization, Imaging, and Image Processing*. (2005) 343–348
12. Tompkins, W.J., ed.: *Biomedical Digital Signal Processing*. Prentice-Hall International, Inc. (1993)

# Nucleus Classification and Recognition of Uterine Cervical Pap-Smears Using FCM Clustering Algorithm

Kwang-Baek Kim<sup>1</sup>, Sungshin Kim<sup>2</sup>, and Gwang-Ha Kim<sup>3</sup>

<sup>1</sup> Dept. of Computer Engineering, Silla University, Busan, Korea  
gbkim@silla.ac.kr

<sup>2</sup> School of Electrical Engineering, Pusan National University, Busan, Korea  
sskim@pusan.ac.kr

<sup>3</sup> Dept. of Internal Medicine, Pusan National University College of Medicine, Busan, Korea  
doc0224@chol.com

**Abstract.** Segmentation for the region of nucleus in the image of uterine cervical cytodiagnosis is known as the most difficult and important part in the automatic cervical cancer recognition system. In this paper, the nucleus region is extracted from an image of uterine cervical cytodiagnosis using the HSI model. The characteristics of the nucleus are extracted from the analysis of morphometric features, densitometric features, colormetric features, and textural features based on the detected region of nucleus area. The classification criterion of a nucleus is defined according to the standard categories of the Bethesda system. The fuzzy *c*-means clustering algorithm is employed to the extracted nucleus and the results show that the proposed method is efficient in nucleus recognition and uterine cervical Pap-Smears extraction.

## 1 Introduction

Cervical cancer is one of the most frequently found diseases in Korean women but can be cured if detected early enough. Conquering the disease is a very important matter. Previous research shows that cervical cancer occupies 16.4 ~ 49.6% of malignant tumors in Korea and occupies 26.3 ~ 68.2% of malignant tumors in women [1], [2]. The best method to completely cure cervical cancer is to prevent the cell from developing into cervical cancer. For this purpose, there have been many efforts to completely or at least partially automate the process of cytodiagnosis during the last 40 years [3].

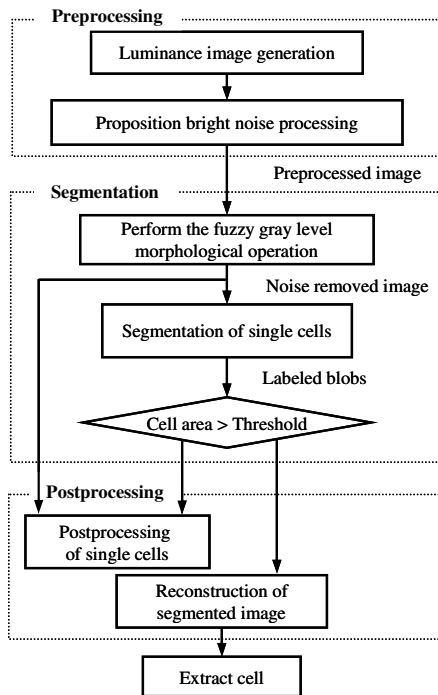
Diagnosis of the region of interest in a medical image comprises area segmentation, feature extraction and characteristic analysis. In area segmentation, a medical specialist detects abnormal regions of a medical image based on his expertise. In feature extraction, features are extracted from the separated abnormal region. A medical doctor diagnoses a disease by using character analysis which is deciphering the extracted features to analyze and compare clinical information. Area segmentation methods used on a nucleus differs according to the target image. The approaches can be largely divided into the pixel-center method and the area-center method [4], [5]. A pixel-center method assigns an independent meaning to each pixel according to a predefined criterion. Pixel-center methods can use the overall characteristic [4].

Although the area-center method needs relatively more calculation time than the pixel-center method, it provides the usage of regional characteristics [5].

In this paper, the following simplification process allows the nucleus to be more easily detected in the image: (i) converting the extracted image of cervix uteri cytodiagnosis to a grey scaled image, (ii) removing noise using brightness information, and (iii) applying a  $5 \times 5$  fuzzy grey morphology operation. To divide the nucleus area in the simplified image, the dark area of the nucleus is separated. Next, the following characteristic information is extracted: 13 morphometric features, and 8 densitometric features, 18 colorimetric features, and one textural feature. Extracted information is categorized into 4 degrees based on the extent of abnormality in each nucleus by using the fuzzy c-means clustering algorithm.

## 2 Nucleus Area Segmentation of Cervix Uteri Cyodiagnosis

The proposed algorithm for extracting the nucleus of cervix uteri cyodiagnosis is shown in Fig. 1.



**Fig. 1.** Process to extract nucleus of cervix uteri cyodiagnosis

In this paper, noise (leukocyte, etc.) is removed by using the information that the size of noise is smaller than that of the nucleus. Small-sized cervix uteri is normal and can therefore be removed because they do not have an influence on detecting an

abnormal cell [6]. In order to analyze the changes and characteristics of a nucleus, 13 morphometric features, 8 densitometric features, 18 colorimetric features, and a textural feature are extracted after detecting the nucleus in cervix uteri. The classification criterion of extracted nucleus is defined according to the standard categories of the Bethesda system. The fuzzy c-means clustering algorithm is applied to classify the malignancy degree of the extracted nucleus according to the standard criterion.

## 2.1 Noise Exclusion Using HSI Information

In the pre-treatment process, color images are changed into grey images. To improve the quality of the image, noise is removed by using a variation of brightness which can be acquired by p-tile stretching. After noise is removed from the image, it is divided into  $8 \times 8$  blocks, to each of which Eq. (1) is applied to each block.

$$z' = \frac{b' - a'}{b - a} \times (z - a) + a' \quad (1)$$

In Eq. (1),  $a$  and  $b$  are the values of brightness in the original image and  $z$  is  $a \leq z \leq b$ . Here,  $a$  is the lowest brightness value + (30% of the highest brightness value) and  $b$  is 80% of the highest brightness value.  $a'$  and  $b'$  are 0 and 255 respectively. An image of cervix uteri cytodiagnosis that is used in this paper is shown in Fig. 2(a). The grey image in which noise is removed by the proposed pre-treatment process is in Fig. 2(b).



**Fig. 2.** Image of cervix uteri cytodiagnosis with noise reduction: (a) Image of cervix uteri cytodiagnosis, (b) Image of cervix uteri cytodiagnosis which noise is removed

## 2.2 Morphology Operation Using Fuzzy

If noise is removed by the proposed pre-treatment process, partial information of the normal cell nucleus and cancer cell nucleus is lost, making it difficult to precisely extract the nucleus. Therefore, by applying a  $5 \times 5$  fuzzy grey morphology operation to improve this problem, the extracted nucleus of the normal cell and abnormal cell can be precisely extracted. Fuzzy morphology operation is in Eq. (2) and (3). The resulting image that is applied to a  $5 \times 5$  fuzzy grey Erosion · Dilation morphology operation is shown in Fig. 3. Here,  $a$  is the original image and  $b$  is the  $5 \times 5$  mask.

$$A \odot B = \{(x, \mu_{A \odot B}(x)) \mid x \in E^N\},$$

$$\mu_{A \odot B}(x) = \inf_{z \in E^N} \min [1, MRF(\mu_A(z), \mu_{(B;x)}(z))],$$

$$MRF(a, b) = \begin{cases} 0 & \text{if } b = 0 \\ a/b & \text{otherwise,} \end{cases} \tag{2}$$

$$A \oplus B = \{(x, \mu_{A \oplus B}(x)) \mid x \in E^N\},$$

$$\mu_{A \oplus B}(x) = \sup_{z \in E^N} [\mu_A(z) \times \mu_{(B;x)}(z)].$$
(3)

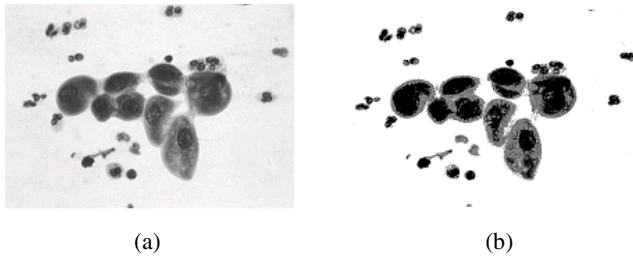


Fig. 3. Images of the closing results: (a) Erosion, (b) Dilation

### 2.3 Area Segmentation of Nucleus Using Repeat Threshold Choice Method

The threshold is chosen by using an iterative threshold selection method in 45% to 100% section of the histogram that is based on the simplified image using the proposed pre-treatment process as shown in Fig. 4. Fig. 5 is the resulting image that comes from dividing the nucleus using the repeat threshold choice method.

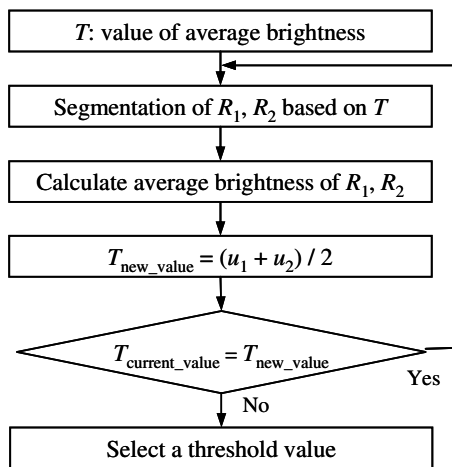


Fig. 4. Algorithm for the selection of the critical value



Fig. 5. Image of segmentation for the region of nucleus

### 2.4 Nucleus Characteristic Extraction for Cancer Cell Recognition

A normal nucleus in cervix uteri cytodiagnosis appears small and pale and the nucleus-cytoplasm ratio is also small. On the other hand, an abnormal cell nucleus has a large size and longish or irregular shape compared to a normal cell. Also, because the dyeing rate is different in an abnormal cell, an abnormal nucleus appears dark and the chromatin in the nucleus does not appear even but rough [7], [8].

In this paper, to classify these characteristics, the characteristics of the nucleus and cell image were extracted. First, the following features are extracted for nucleus characteristic: area of nucleus, circumference of nucleus, ratio between circumference of nucleus and circumference of quadrilateral, degree of roundness in nucleus' shape, reciprocal of degree of roundness in nucleus' shape, log10 (height/width) in the smallest area of quadrilateral, the longest interior line in horizontal and vertical directions, ratio between area of nucleus and area of quadrilateral. Second, we calculate the area that includes the exterior area of the nucleus and the area of the convex hull wrapped around the nucleus in the convex range.

In the information about brightness, pixels with the following characteristics in which the values are over 60 are extracted: average value, standard deviation, dispersion of brightness, histogram's maximum light and darkness value, central value, smallest light and darkness value, brightness value. Information about color is calculated using the average and standard deviation from the components of red, green, blue, hue, saturation, and intensity. HVS divides texture information into channels in which the energy vector and energy deviation are calculated. Texture characteristic vector that uses energy is calculated by the following.

$$q_{mn} = C_{mn} \sum_{\omega} \sum_{\theta} [p_{\theta}(\omega)]^2, \tag{4}$$

$$e_{mn} = \log(1 + p_{mn}). \tag{5}$$

Here  $p_{\theta}(\omega)$  represents the value in frequency space of each channel.  $C_{mn}$  is a constant for the normalization value. The calculation of the texture feature using energy deflection is as follows [9]:

$$q_{mn} = \sqrt{D_{mn} \sum_w \sum_{\theta} [(p_{\theta}(w))^2 - p_{mn}]^2}, \tag{6}$$

$$d_{mn} = \log(1 + q_{mn}). \tag{7}$$

Here,  $D_{mn}$  is a constant for the normalization value. The calculated values from equation (4) to (7) and the texture representation that displays the texture feature of a nucleus using the average value and standard deviation of an image is expressed in Eq. (8).

$$Descriptor_{texture} = \begin{bmatrix} dc & std & e_{00} & e_{01} & \dots \\ e_{45} & d_{00} & d_{01} & \dots & d_{45} \end{bmatrix}. \tag{8}$$

### 2.5 Nucleus Classification and Recognition Using FCM Clustering Algorithm

Basically, the fuzzy c-means clustering algorithm resembles the K-means clustering algorithm in that it divides each cluster by using the Euclidian distance of a data set [10]. The only difference is that it does not present whether data belongs to each cluster (0 or 1) but presents a degree (a real number between 0 and 1) of belonging with classification matrix. Also the beginning point is defined in the middle of the whole date by classification matrix that is made temporarily when the beginning point is set. Fuzzy c-means clustering algorithm that is used in this paper is shown in Fig. 6.

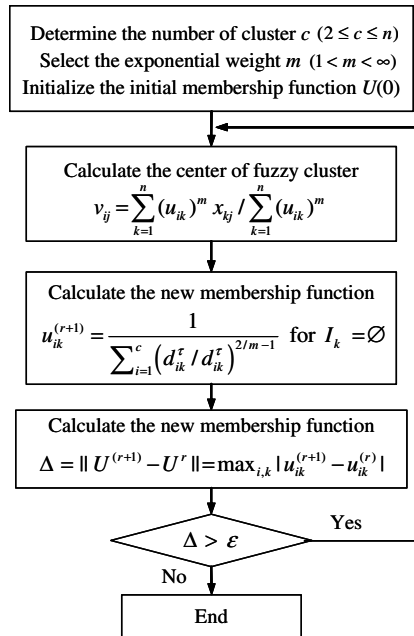


Fig. 6. Fuzzy c-means clustering algorithm

## 3 Experiment and Result Analysis

The environment of the experiment is embodied by Visual C++ 6.0 and C++ Builder 6.0 in Pentium-IV PC of the IBM compatible. The specimen was 20 samples of



640\*480 cervix uteri cytodiagnosis image size and it was acquired from Pusan university hospital. The analysis result of a cervix uteri cytodiagnosis image with the proposed method is shown in Fig. 7 (a), and the nucleus extraction result of the cervix uteri cytodiagnosis is shown in Fig. 7 (b).

To evaluate the performance of the proposed nucleus segmentation method on 20 actual images, the method was compared with the diagnostic results of a medical specialist. The nucleus number of cervix uteri cytodiagnosis extracted from a medical specialist in 20 samples is 316, and the number extracted from this research is 278. Table 1 displays the number of nucleus that was extracted by the proposed method.

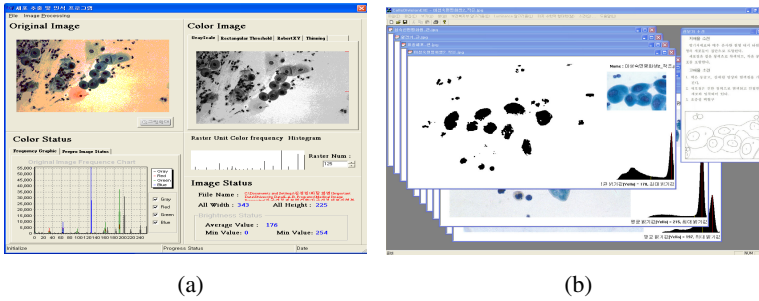


Fig. 7. Nucleus extraction result of the cervix uteri cytodiagnosis: (a) Image analysis of proposed cervix uteri cytodiagnosis, (b) result of nucleus extraction of cervix uteri cytodiagnosis

Table 1. Nucleus extraction result

	Medical Specialist	Proposed method	Extraction rate of the proposed method
Nucleus extraction	316	278	87.9%

As can be concluded from Table 1, the accuracy of extraction rate is 87.9% in this research. When two or more cluster cells piled up, extraction did not occur correctly. A nucleus can be classified into normal cell nucleus, abnormal cell nucleus and cancer cell nucleus. Therefore, in this paper, the nucleus is divided into the following 5 classes based on the Bethesda System: WNL, ACUS, LSIL, HSIL, SCC. Here, WNL is a normal cell but as it progresses, it has high malignancy in an abnormal cell. Cells that are classified as SCC are cancer cells. Fig. 8 displays nucleus characteristic information of cervix uteri cytodiagnosis that was extracted using the proposed method in this paper.

The classification of nucleus characteristic information based on the Bethesda System by using nucleus characteristic information such as in Fig. 8 is in Fig. 9. By using standards of nucleus classification information to identify normal cells and cancer cells such as in Fig. 9, the accuracy of cancer diagnosis can be much more improved compared to classifying the whole nucleus that appears in the image.

In Fig. 8, fuzzy c-means clustering algorithm is applied to classify and distinguish the characteristic information of the extracted nucleus, normal cell, abnormal cell and cancerous cell. The results from classifying and distinguishing the state of the cell using final FCM clustering algorithm is in Table 2.

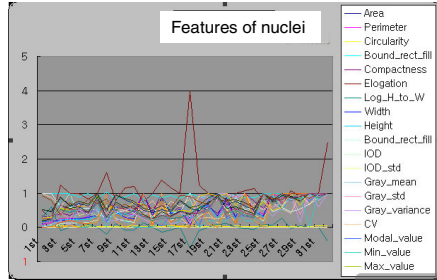


Fig. 8. Characteristic information of nuclei

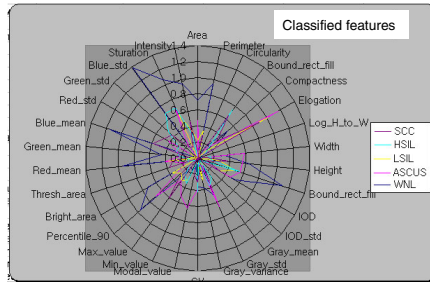


Fig. 9. Information of the standard category in the Bethesda system

Table 2. Classification and recognition results of cell by FCM clustering algorithm

Medical specialist		Proposed method	
Normal cell(WNL)	88	Normal cell(WNL)	68
	ASCUS 78	ASCUS	90
Abnormal cell	LSIL 40	Abnormal cell	LSIL 45
	HSIL 51		HSIL 54
Cancer cell(SCC)	21	Cancer cell(SCC)	21

Listing in order, the 5 classes WNL, ACUS, LSIL, HSIL, SCC is divided by the Bethesda System, 5 clusters were set for classification as can be seen in Table 2. As can be concluded from Table 2, much more frequently in the proposed method were normal cells classified as abnormal cells than in the diagnosis of a medical specialist. This is because the extracted nucleus is dyed, making it open to being classified as an

abnormal cell. But, it can be confirmed that the accuracy between abnormal cells and cancer cells, which a medical specialist diagnosis has little performance however the classification grade of abnormal cells differs. The reason is that FCM clustering algorithm does not sort correctly the dyeing density of nucleus information, which is partial information of the abnormal nucleus cell. But it can be concluded through Table. 2 that the proposed method is comparatively efficient in classifying abnormal cells and cancer cells, and can help a medical specialist in diagnosis.

## 4 Conclusion

Because cervix uteri cytodiagnosis is various and complicated, it is difficult to extract and identify cell nucleus efficiently with existing image processing methods.

In this paper, the extraction and identification method of cervix uteri cytodiagnosis using 5 classes WNL, ACUS, LSIL, HSIL, SCC based on the Bethesda System was proposed. In this paper, the nucleus is easily detected by using the following simplification process of the image: (i) converting the extracted image of cervix uteri cytodiagnosis to a grey-scaled image, (ii) removing noise using brightness information, and (iii) applying a  $5 \times 5$  fuzzy grey morphology operation. To segment the nucleus area in the simplified image, the dark area of the nucleus is separated, and then the following characteristic information is extracted: 13 morphometric features, and 8 densitometric features, 18 colorimetric features, and a textural feature. Extracted information in each nucleus is categorized and recognized into normal cells, abnormal cells with 3 degrees, and cancer cells by the fuzzy c-means clustering algorithm.

As a result of experimenting with 20 samples of cervix uteri cytodiagnosis, it can be confirmed that the proposed method is efficient in recognizing abnormal cells and cancer cells and has little difference with the diagnosis of a medical specialist.

In the future, studies must be conducted to correctly extract characteristic information of nucleus by analyzing morphology and color characteristics of the nucleus and by making a standard of classification to reduce presumption error in nucleus classification by extracting much information.

## References

1. Rutenberg, Mark R.: Neural Network Based Automated Cytological Specimen Classification Systems and Method. United States Patent, Patent No.4965725 (1990)
2. Grohs, Heinz K., Husain, O. A. Nassem: Automated Cervical Cancer Screening. Igakushoin. (1994)
3. Seo, C. W., Choi, S. J., Hong, M. K., Lee, H. Y., Jeong, W. G.: Epidemiologic Observation of Diagnosis of Cervical Cancer and Comparative Study of Abnormal Cytologic Smear and Biopsy Result. Journal of The Korean Academy of Family Medicine. Vol. 17, No. 1. (1996) 76-82
4. Banda-Gamboa, H, Ricketts, I, Cairns, A, Hussein, K, Tucker, JH, Husain, N.: Automation in cervical cytology: an overview. Analytical Cellular Pathology, Vol. 4. (1992) 25-48
5. Lee, J. D.: Color Atlas Diagnostic Cytology: Press of Korea Medical Publishing Company. (1989)

6. Kim, H. Y., Kim, S. A., Choi, Y. C., Kim, B. S., Kim, H. S., Nam, S. E.: A Study on Nucleus Segmentation of Uterine Cervical Pap-Smears using Multi Stage Segmentation Technique. *Journal of Korea Society of Medical Informatics*. Vol.5, No.1. (1999) 89-95
7. Kim, K. B., Yun, H. W.: A Study on Recognition of Bronchogenic Cancer Cell Image Using a New Physiological Fuzzy Neural Networks. *Japanese Journal of Medical Electronics and Biological Engineering*. Vol.13, No.5. (1999) 39-43
8. Bishop, C. M.: *Neural Networks for Pattern Recognition*, Oxford University Press. (1995)
9. Manjunath, B. S., Ma, W. Y.: Texture Features for Browsing and Retrieval of Image Data. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8. (1996)
10. Klir, George J., Yuan, Bo: *Fuzzy Sets and Fuzzy Logic Theory and Applications*. Prentice Hall PTR (1995)

# Rib Suppression for Enhancing Frontal Chest Radiographs Using Independent Component Analysis

Bilal Ahmed<sup>1</sup>, Tahir Rasheed<sup>1</sup>, Mohammed A.U. Khan<sup>2</sup>, Seong Jin Cho<sup>1</sup>,  
Sungyoung Lee<sup>1</sup>, and Tae-Seong Kim<sup>3,\*</sup>

<sup>1</sup> Department of Computer Engineering, Kyung Hee University, South Korea  
{bilal, tahir, sjcho, sylee}@oslab.khu.ac.kr

<sup>2</sup> Comsats Institute of Technology, Abbotabad, Pakistan  
mohammad\_a\_khan@yahoo.com

<sup>3</sup> Department of Biomedical Engineering, Kyung Hee University, South Korea  
tskim@khu.ac.kr

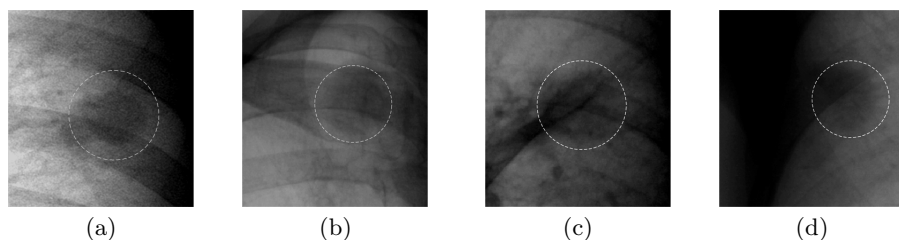
**Abstract.** Chest radiographs play an important role in the diagnosis of lung cancer. Detection of pulmonary nodules in chest radiographs forms the basis of early detection. Due to its sparse bone structure and overlapping of the nodule with ribs and clavicles the nodule is hard to detect in conventional chest radiographs. We present a technique based on Independent Component Analysis (ICA) for the suppression of posterior ribs and clavicles which will enhance the visibility of the nodule and aid the radiologist in the diagnosis process.

## 1 Introduction

Chest X-rays play an important role in the diagnosis of lung cancer. According to the latest statistics provided by the American Cancer Society, lung cancer is estimated to produce 174,470 new cases in 2006, accounting for about 12 percent of all cancer diagnoses. Lung cancer is the most common cancer related death in both men and women. An estimated 162,460 deaths, accounting for about 29 percent of all cancer related deaths, are expected to occur in 2006. Early detection of lung cancer is the most promising strategy to enhance a patients' chances of survival. Early detection can be achieved in a population screening: the most common screenings for lung cancer make use of Chest Radiography, or low radiation dose Computer Tomography (CT) scans. Despite the development of advanced radiological exams such as CT the conventional Chest X-ray remains the most common tool for the diagnosis of lung cancer. The main reason behind this being the fact that CT and helical CT exams expose the patient to a higher dose of radiation, estimated to be about 100 times higher than that for a conventional chest X-ray.

---

\* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)(IITA-2006-(C1090-0602-0002)).



**Fig. 1.** Effect of location on nodule detection: (a) Nodule with overlapping posterior and anterior ribs. (b) Overlapping clavicle. (c) Overlapping rib and blood vessels. (d) Nodule lying in the hilum area of the radiograph. (Contrast enhanced images).

Due to the heavy use of conventional chest x-rays for early detection, there is a need for enhancing its diagnostic value. In a recent study carried out to assess various reasons for litigation against physicians, it was revealed that failure to diagnose lung cancer accounted for 80 percent of the cases [13]. For these reasons, there has been a particular interest for the development of computer aided diagnostic (CAD) systems that can serve as a follow-up reader, paying attention to the suspicious regions in the radiograph that then have to be examined by a radiologist. Earlier work on CAD systems for automated nodule detection in chest radiographs was reported in [7]. The process for nodule detection employed multiple gray-level thresholding of the difference image (which corresponds to the subtraction of a nodule-enhanced image and a nodule-suppressed image) and then classification. The system resulted in a large number of false positives which were eliminated by adaptive rule-based tests and an artificial neural network (ANN). Deus Technologies received FDA pre-market approval for its RapidScreen CAD system in July 2001. Its intended use is to identify and mark regions of interest on digital or digitized frontal chest radiographs. The authors in [8] evaluated the usefulness of CAD that incorporated temporal subtraction for the detection of solitary pulmonary nodules on chest radiographs. The authors in [9] concluded that the accuracy of radiologists in the detection of some extremely subtle solitary pulmonary nodules can be improved significantly when the sensitivity of a computer-aided diagnosis scheme can be made extremely high. The authors noted however, that all of the six radiologists failed to identify some nodules (about 10 percent), even with the correct computer output.

The analysis of existing CAD systems revealed that the detectability of the nodule is largely dependent on its location in the chest radiograph. The problematic areas for nodule detection can be categorized as follows: 1) a nodule overlaps completely with an anterior rib and partly with the posterior rib, as shown in Fig. 1. The ribs need to be suppressed for identifying the true shape of the nodule; 2) a nodule hidden in the hilum area (the area surrounding the lung cavity which has high illumination) is hard to detect due to poor contrast as shown in Fig. 1; 3) nodule overlaps with the clavicles as shown in Fig. 1. Therefore, the suppression of ribs and clavicles in chest radiographs would be potentially

useful for improving the detection accuracy of a given CAD system. An attempt was made in [2] where the rib cage was suppressed by means of Massive Training Artificial Neural Network (MTANN). A drawback of the system is the need of a dual-energy bone image in which the ribs are separated from the soft-tissue at the initial training phase. The technique was found to be sensitive to the noise levels, due to the subtraction process.

From a pattern recognition point of view, the rib-cage can be considered as a separate structure that needs to be suppressed independently from the rest of the radiograph. We formulate the problem of chest radiograph enhancement considering rib-cage as one class and rest of the image as another separate class. In order to discriminate between the rib-cage class and the other we need data-dependant basis functions that naturally align themselves with the rib structure and remain independent from the other class.

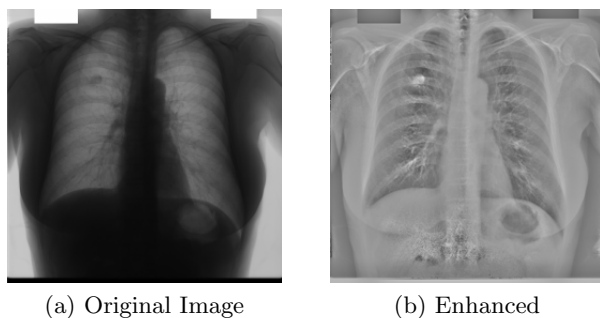
Independent Component Analysis (ICA) has been used quite recently for enhancing EEG signals [10]. It has been shown that ICA can remove undesired artifacts from the EEG signal without causing any substantial damage to the original source signal. The main motivation behind such applications of ICA is the reasonable assumption that noise unrelated to the source is independent from the event-related signals, and hence can be separated. For image data ICA basis work as localized edge filters [5]. In a chest radiograph the ribcage constitutes a sparse bony structure and thus contributing significantly to the overall edge count. ICA can be used to find basis functions which represent this structure and its appropriate characteristics. Our work includes the creation of ICA basis from an enhanced chest radiograph, clustering these basis and the reconstruction of the chest radiograph with the help of non-edge basis. These non-edge basis would correspond to the non-rib component and the exclusion of edge basis would suppress the bone-structure and other undesired edges in the original image.

## 2 Image Enhancement

Chest radiographs contain non-uniform illumination due to the acquisition apparatus and their complex structure. Non-uniform illumination can distort edges of ribs and suppress the detail of the texture. Due to this non-uniform illumination pattern the nodule might get partially obscured and lose its basic characteristics. The ribs and the blood vessels can be viewed as potential edges, and in order to clearly demarcate them it is desirable that they have high detail in the image. Removing the non-uniform illumination pattern and making the image homogeneous would enhance the texture detail and the rib edges.

Conventional methods of homogenizing image such as homomorphic filtering assume an illumination and reflectance model which considers the illumination as a multiplicative component. We have adopted the technique of [1] for normalizing the image's intensities locally. The technique for normalization achieves results which enhance the overall image contrast and strengthen the edges.

$$I_N = \frac{I - I_{LP}}{\sqrt{(I^2)_{LP} - (I_{LP})^2}} \quad (1)$$



**Fig. 2.** Enhancing the original chest radiograph

where 'LP' simply denotes Gaussian blurring of appropriate scale. This local normalization can be considered as making the image zero mean and unit variance. Figure 2 shows the result of this local normalization on a conventional chest radiograph.

### 3 Independent Component Analysis

ICA performs a blind source separation (BSS), assuming linear mixing of the sources. ICA generally uses techniques involving higher-order statistics. Several different implementations of ICA can be found in the literature [4]. We will not discuss those implementations here and restrict ourselves to the topic. Let us denote the time varying observed signal by

$$\mathbf{x} = (x_1 x_2 \dots x_n)^T$$

and the source signal consisting of independent components by

$$\mathbf{s} = (s_1 s_2 \dots s_m)^T$$

The linear ICA assumes that the signal  $x$  is a linear mixture of the independent components,

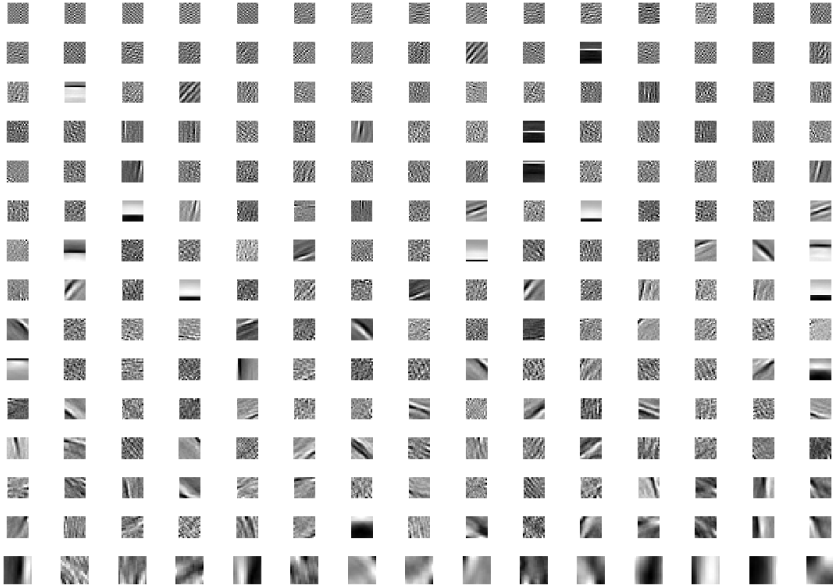
$$\mathbf{x} = \mathbf{A} \cdot \mathbf{s} \tag{2}$$

where the matrix  $\mathbf{A}$  of size  $m \times n$  represents linear memory less mixing channels. It is often assumed that  $n = m$  (complete ICA) for simplicity, which can be relaxed without any loss of generality. A common preprocessing step is to zero the mean of the data and then apply a linear "whitening" transform so that the data has unit variance and is uncorrelated. The algorithms must find a separating or de-mixing matrix such that

$$\mathbf{s} = \mathbf{W} \cdot \mathbf{x} \tag{3}$$

where ( $\mathbf{W}$ ) is the de-mixing matrix. ICA has some limitations which can be summarized as: 1) neither energies nor signs of the ICs can be calculated;





**Fig. 3.** ICA basis for a chest radiograph

2) there is no ordering of the found ICs. Higher-order statistics have been adopted to estimate independent sources in ICA, like kurtosis (the fourth order cumulant) and the negentropy.

### 3.1 Application to Chest Radiographs

Chest radiographs can be viewed as a mixture of two components the bone-structure and the soft-tissue. From an image processing point of view the first component is mostly composed of edges (due to the sparse bony structure). Suppressing these edges, without affecting the other information in the radiograph would enhance the remaining information pertaining to the second component.

ICA features for natural images have been described in literature as being Gabor filters [3][4]. Application of ICA on medical images such as MRI scans of the skull showed that ICA produced basis which could be more efficiently described as step functions, mainly due to the presence of a large number of flat intensity values with rapid transitions [3]. In our case the complete set of basis found for a single nodule-containing radiograph are shown in Fig. 3. As can be seen the basis are a combination of Gabor filters and step functions (depicting the rapid transition of intensity values). These basis can be best described as being edge filters [5]. Analysis of these basis reveal the fact that there are inherently three major classes in the chest radiograph: namely, rib-cage and blood vessel edges, noise, and the background (containing the hilum, organ shadows and soft-tissue).

Image reconstruction using only a subset of these basis would result in an image in which the all the information related to the left-out basis would be suppressed. For this specific application we would like to leave out the basis that contain the information pertaining mainly to the edges. Simple unsupervised clustering methods such as the k-means clustering algorithm can be employed for the classification of these basis into their respective classes. A distance or similarity metric is needed for describing each class. This distance metric can be obtained from simple kurtosis values of the obtained basis. Image reconstruction from the selected basis would involve making the other basis vectors zero. Let  $C_1$  and  $C_2$  be two classes obtained and the corresponding basis matrix is  $\mathbf{A}_1$  and  $\mathbf{A}_2$  having the same dimensions as that of the original basis matrix  $\mathbf{A}$ , but containing zero vectors in place of non-class basis. The image reconstruction for  $n$  classes can be given as,

$$\mathbf{x}_i = \mathbf{A}_i \cdot \mathbf{s} \quad i = 1, 2, \dots, n \quad (4)$$

thus obtaining a set of  $n$  images corresponding to the  $n$  classes.

## 4 Experiment and Results

The chest radiographs were taken from the JSRT database [11]. The images are digitized to 12 bits posterior-anterior chest radiographs, scanned at a resolution of 2048 x 2048 pixels; the size of one pixel is 0.175 x 0.175 mm<sup>2</sup>. The database contains 93 normal cases and 154 cases of proven lung nodule. Diameters and the positions of the nodules are provided along with the images. The nodule diameters range from 5-60 mm, and are located throughout the lung field, and their intensities vary from nearly invisible to very bright. The nodules present in the database are representatives of the problems we delineated in the introduction.

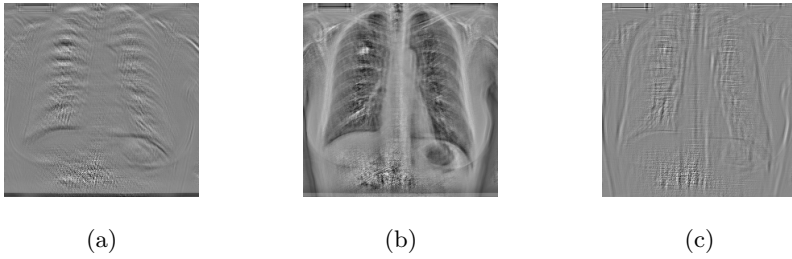
### 4.1 Preprocessing

The images were first enhanced according to the method discussed in Sect. 2. The images were down-sampled from 2048 x 2048 to 256 x 256 pixels. Overlapping 15 x 15 blocks of the image were taken as the input to the ICA algorithm.

### 4.2 Results

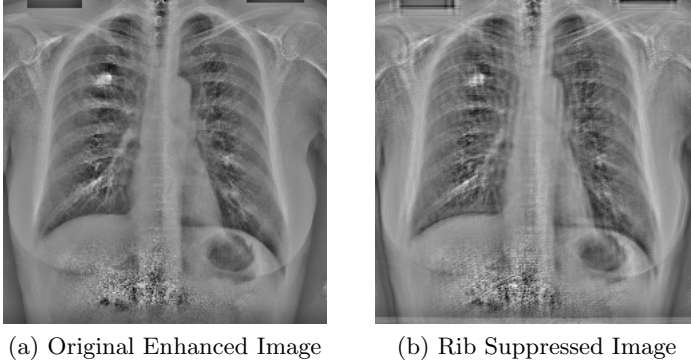
The data was initially made zero mean and whitened. The FastICA [4] algorithm was used with the tanh(hyperbolic tangent) non-linearity relating to the Super-Gaussian source distributions. Note that we did not apply PCA reduction at any stage to preserve the texture of the overall chest radiograph. Next we show the ICA basis functions obtained and discuss their relevance to our chest radiograph analysis.

Figure 3 shows the resulting basis vectors– (i.e., columns of the  $\mathbf{A}$  matrix). FastICA was used for learning the structure inherently present in the candidate image. The structure contained in the blocks can be best analyzed by inspecting



**Fig. 4.** Three classes generated from the clustering of ICA basis.(a) and (b) depict the images corresponding to the bone and noise components whereas (c) depicts the image corresponding to the non-edge basis.

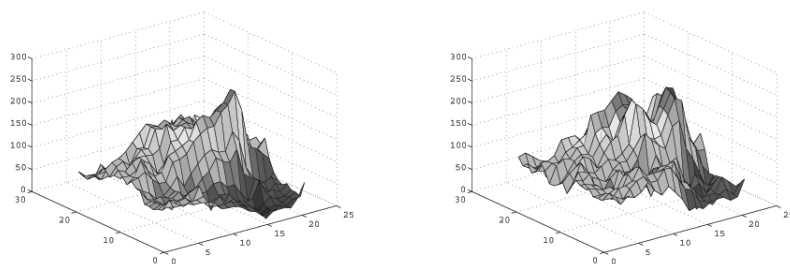
the linear basis extracted through the algorithm. These basis are used to reconstruct the original image along with the help of the independent components. As can be seen, the resulting basis vectors are localized, oriented and have multiple scales. Second, most of the ICA basis vectors include a small global step-like grayscale change. However, majority of the basis vectors can be conveniently associated with the directional feature associated closely with the ribcage. We do find some small high frequency edges in the basis that seem to be linked with small tissue like structure in the chest radiograph.



**Fig. 5.** Effect of rib-suppression, the ribs are suppressed but the nodule remains intact

The basis so obtained were then clustered according to the k-means algorithm with the clustering feature being the kurtosis of the basis images. The results of the clustering can be seen from Fig. 4. The final rib-suppressed image is shown in Fig. 5 along with the original enhanced image initially given to ICA. As can be seen that other structures as well as the ribs have been suppressed, with the original nodule left intact in the due course.

The effect of the suppression from a pattern recognition point of view can be characterized as increasing the Gaussianity of the nodule. Pulmonary chest



(a) Before Rib Suppression,  
 $kurtosis = -1.18$

(b) After Rib Suppression,  
 $kurtosis = -0.74$

**Fig. 6.** Gaussianity of the nodule before and after rib-suppression. (*Excess Kurtosis is shown( $kurtosis-3$ )*).

nodules have been characterized as Gaussian shapes in [1]. The overlapping of the ribs and other bony structure tends to distort this Gaussianity as can be seen from Fig. 6. The removal of these overlapping structures increases the Gaussianity of the underlying nodule. Simple blob detection techniques [6] can then be employed for the detection of these nodules with more accuracy [1]. An illustration of the enhanced Gaussianity is given in Fig. 6.

## 5 Conclusion and Future Work

We have demonstrated that the suppression of ribs and clavicles can be efficiently achieved through the use of Independent Component Analysis. The removal of these artifacts result in the enhancement of the nodule. This increase in the overall Gaussianity of the nodule would enable a standard multi-scale blob-detector [6] to detect a nodule with more accuracy [1].

In our future work, we would like to apply constrained-ICA [12] for the suppression of ribs and removing the blood vessels which hinder the efficient detection of nodules. We would also like to assess the performance of a standard classifier for obtaining the number of false positives produced as a result of applying ICA for radiograph enhancement.

## References

1. Schilham, A.M.R., Ginneken, B.V., Loog, M.: A computer-aided diagnosis system for detection of lung nodules in chest radiographs with an evaluation on a public database. *Medical Image Analysis*, **10** Elsevier (2006) 247–258
2. Suzuki, K., Abe, H., MacMahon, H., Doi, K.: Image-Processing Technique for Suppressing Ribs in Chest Radiographs by Means of Massive Training Artificial Neural Network (MTANN). *IEEE transactions on Medical Imaging* **25**(4) (2006)
3. Inki, M.: ICA features of image data in one, two and three dimensions. *Proceedings of Independent Component Analysis (ICA)-2003*. IEEE (2003) 861–866

4. Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley Interscience (2001)
5. Bell, A.J., Sejnowski, T.J.: The 'Independent Components' of natural images are edge filters. *Vision Research* (1997) 3327–3338
6. Lindeberg, T.: Feature Detection with automatic scale selection. *International Journal of Computer Vision*, **30** (1998) 79–116
7. Xu, X.W., Doi, K., Kobayashi, T., MacMahon, H., Giger, M.L.: Development of an improved CAD scheme for automated detection of lung nodules in digital chest images. *Med Phys.* **24** (1997) 1395–1403
8. Johkoh, T., Kozuka, T., Tomiyama, N., Hamada, S., Honda, O., Mihara, N., Koyama, M.: Temporal subtraction for detection of solitary pulmonary nodules on chest radiographs: evaluation of a commercially available computer-aided diagnosis system. *Radiology* **223** (2002) 806–811
9. Shiraishi, J., Abe, H., Engelmann, R., Doi, K.: Effect of high sensitivity in a computerized scheme for detecting extremely subtle solitary pulmonary nodules in chest radiographs: observer performance study. *Acad. Radiology* **10** (2003) 1302–1311
10. Lange, D.H., Pratt, H., Inbar, G.F.: Modeling and Estimation of single evoked brain potential components. *IEEE transactions on BioMedical Engineering* **44** (1997)
11. Shiraishi, J., Katsuragawa, S., Ikezoe, J., Matsumoto, T., Kobayashi T., Komatsu, K., Matsui, M., Fujita, H., Kodera, Y., Doi, K.: Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. *American Journal of Roentgenology* **174** (2000)
12. Lu, W., Rajapakse, J.C.: Approach and Applications of Constrained ICA. *IEEE transactions on Neural Networks*. **16**(1) (2005)
13. McLean, T.R.: Why Do Physicians Who Treat Lung Cancer Get Sued? *Chest* **126** American College of Chest Physicians (2004)

# A Novel Hand-Based Personal Identification Approach

Miao Qi<sup>1,2</sup>, Yinghua Lu<sup>1</sup>, Hongzhi Li<sup>1,2</sup>, Rujuan Wang<sup>1</sup>, and Jun Kong<sup>1,2,\*</sup>

<sup>1</sup> Computer School, Northeast Normal University, Changchun, Jilin Province, China

<sup>2</sup> Key Laboratory for Applied Statistics of MOE, China

{qim801, luyh, lih857, kongjun}@nenu.edu.cn

**Abstract.** Hand-based personal identification is a stable and reliable biometrically technique in the field of personal identity recognition. In this paper, both hand shape and palmprint texture features are extracted to facilitate a coarse-to-fine dynamic identification task. The wavelet zero-crossing method is first used to extract hand shape features to guide the fast selection of a small set of similar candidates from the database. Then, a circular Gabor filter, which is robust against brightness, and modified Zernike moments methods are used to extract the features of palmprint. And one-class-one-network (Back-Propagation Neural Network (BPNN) classification structure is employed for final classification. The experimental results show the effectiveness and accuracy of the proposed approach.

## 1 Introduction

Recently, automatic biometric systems based on human characteristics for personal identification have been widely researched in applications of personal identity recognition due to the uniqueness, reliability and stability of biometric feature. So far, fingerprint, face and iris recognition have been studied extensively, which result in successful development of biometric systems for commercial applications. However, limited study has been reported on handprint identification and verification. Most of existing hand-based personal identity recognition researches are focus on palmprint recognition. There are mainly two popular approaches for palmprint recognition. The first approach is based on structural features such as principle line [1], [2] and feature point [3] approaches. Although the structural features can represent individual well, they are difficult to extract and need high computation cost for matching. The other approach is based on statistical features which are the most intensively studied and used in the field of feature extraction and pattern recognition, such as Gabor filters [4], [5], eigenpalm [6], fisherpalm [7], texture energy [8], [9], invariant moments [10], Fourier transform [11] and wavelet transform [12]. Statistical features based palmprint is straightforward in that the palmprint image is treated as a whole for extraction in the recognition +process.

---

\* Corresponding author.

This work is supported by science foundation for young teachers of Northeast Normal University, No. 20061002, China.

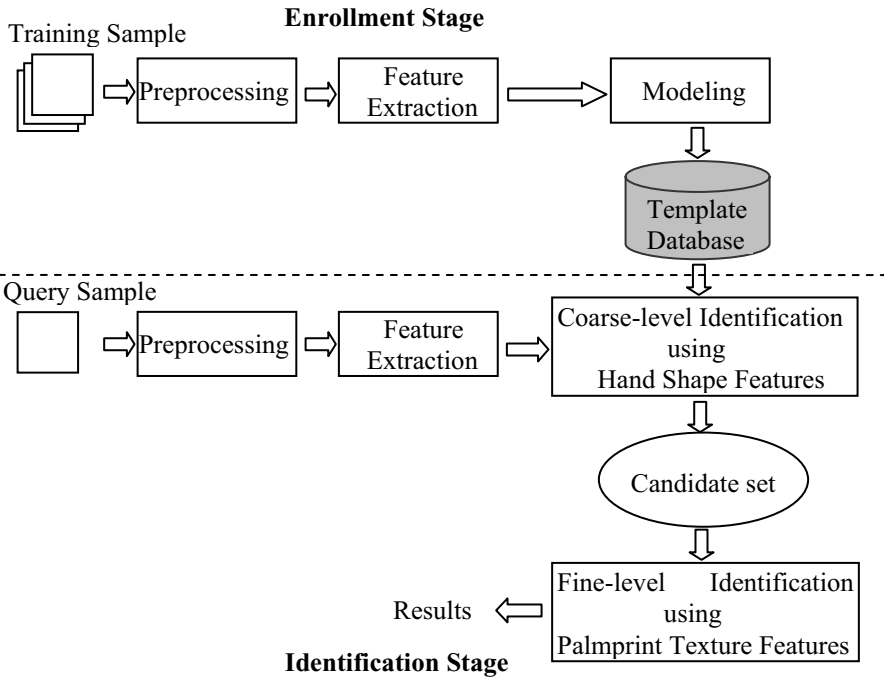


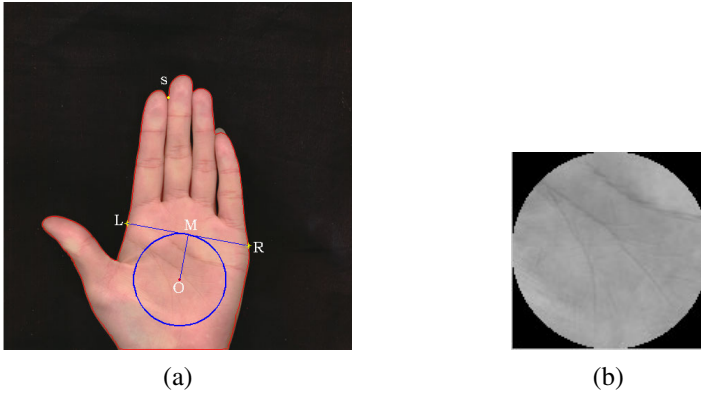
Fig. 1. The flow chart of the identification process

The flow chart of the proposed system is shown in Fig. 1. Our handprint identification system can be divided into two stages, enrollment and identification. Given a query sample, the hand shape features are first extracted by wavelet zero-crossing method to guide to select a small set of similar candidates from the database. Then, statistical features of the palmprint are extracted for determining the final identification from the selected set of similar candidates.

The rest of this paper is organized as follows. Section 2 introduces the handprint image acquisition and the region of interest (ROI) localization. Section 3 describes the feature extraction methods briefly. The process of coarse-to-fine identification strategy is depicted in Sect. 4. The experimental results are reported in Sect. 5. Finally, the conclusions are summarized in Sect. 6.

## 2 Image Acquisition and Preprocessing

A peg-free flatbed scanner is used for handprint image acquisition. By using this device can increase flexibility and friendliness of the system. The users can place their hand freely on the flatbed scanner and only assure that the thumb is separate with the other four fingers, which are incorporated naturally (shown in Fig. 2).



**Fig. 2.** The process of locating ROI

An image threshold method is first proposed to segment the hand image from background. The proposed method can detect fingernails by analyzing the hand color components:  $r$ ,  $g$  which represent red and green, respectively. The proposed method is described as follows:

$$f^*(x, y) = \begin{cases} 0 & r - g < T \\ 1 & \text{otherwise,} \end{cases} \quad (1)$$

Where  $T$  is a threshold used to filter the fingernails and segment the hand from background. Then a contour tracing algorithm starting from the left top point of binary image in the clockwise direction is used to obtain the hand shape contour, whose pixels are recorded into a vector  $V$ . Figure 2 shows that the extracted contour (labelled by red line) of a handprint image which perfectly matches the original hand outline. The point  $S$  is located by chain code method. The distances between  $S$  and  $L$ ,  $S$  and  $R$  are 190 and 240, respectively.  $M$  is the middle point of line  $\overline{LR}$ . The circle region whose center is  $O$  and radius is  $\overline{OM}$  is denoted as ROI (see Fig. 2(b)), where line  $\overline{OM}$  is vertical to line  $\overline{LR}$  and its length is  $3/8$  of the length of line  $\overline{LR}$ .

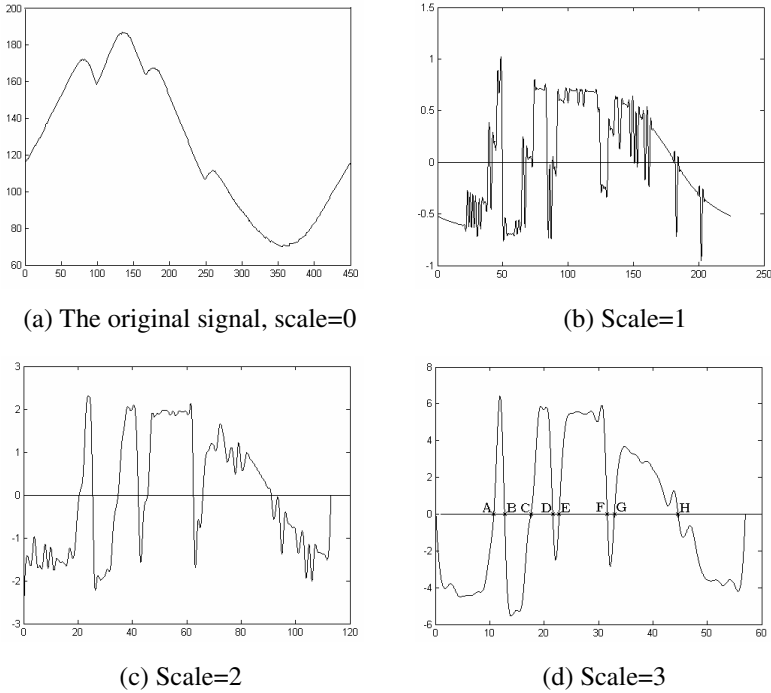
### 3 Feature Extraction

#### 3.1 Hand Shape Extraction Using Wavelet Zero-Crossing

Zero-crossing is a widely and adaptive method used in recognition for object contour. When a signal includes important structures that belong to different scales, it is often helpful to reorganize the signal information into a set of detail components of varying scale. In our work, only the partial contour of hand shape which is stable and includes the dominant difference with individual is used. The length of the partial contour whose first point is at the position 100 before  $S$  in  $V$  is 450 pixels. The 1-D signal (see Fig. 3(a)) is mapped by the distances between contour points and the middle point of the two endpoints of the partial contour. The 1-D signal is transformed by the wavelet



transform. Several high-frequency sub-band signals in various scales are generated as depicted in Fig. 3. When decomposing the 1-D signal, we restrict the dyadic scale to  $2^j$  in order to obtain a complete and stable representation (detailed in [13]) of the partial hand shape contour. The positions of eight zero-crossings (A - H) of at the scale 3 are recorded as features of hand shape (shown in Fig. 3(d)).



**Fig. 3.** (a) The original signal of the partial hand shape contour, scale=0. (b) - (d) The high-frequency sub-bands of the wavelet transformed signal from scales 1 to 3.

### 3.2 Palmprint Feature Extraction

Although the hand shape features are powerful to discriminate many handprints, it can't separate the handprints with similar hand shape contours. Thus, the other features are necessary to be extracted for fine-level identification. By carefully observing the palmprint images, the principal lines and wrinkles can well represent the uniqueness of individual's palmprint. Through comparing in study, we find that the green component which is a main component of an RGB image can describe the texture more clear than the others such as red component and gray image by extracting the texture of ROI using our proposed texture extraction methods.

#### 3.2.1 Circular Gabor Filter

Gabor filter has been already demonstrated to be a powerful tool in the texture analysis, which is very useful in the detection of texture direction. But in rotation invariant

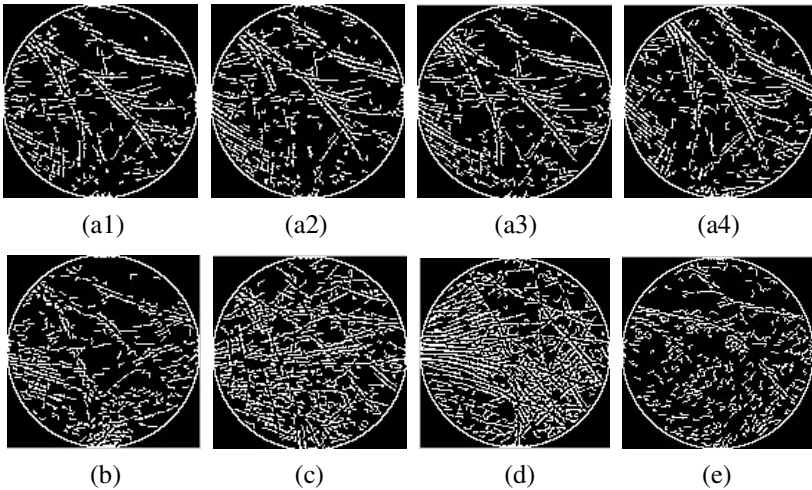
texture analysis, the orientation of texture becomes less important. Thus, traditional Gabor filters are less suitable for this topic. If the sinusoid carries in all orientations, it is circular symmetric (detailed in [14]), which is defined as follows:

$$G(x, y, \sigma, F) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\} \exp\left\{2\pi i F(\sqrt{x^2 + y^2})\right\}, \quad (2)$$

where  $i = \sqrt{-1}$ ,  $F$  is the central frequency,  $\sigma$  is the standard deviation of the Gaussian envelope. In order to provide more robustness to brightness, the Gabor filter is turned to zero DC (Direct Current) with the application of the following formula:

$$G^*(x, y, \sigma, F) = G(x, y, \sigma, F) - \frac{\sum_{i=-n}^n \sum_{j=-n}^n G(x, y, \sigma, F)}{(2n + 1)^2}, \quad (3)$$

where  $(2n + 1)^2$  is the size of filter.



**Fig. 4.** (a1) - (a4) are the filtered ROIs from the same person. (b) - (e) are the filtered ROIs from different persons.

The filtered image is employed by the real part. An appropriate threshold value is selected to binarize the filtered image and morphological operations are applied to remove the spur, isolated pixels and trim some short feature lines and the results are shown in Fig. 4.

### 3.2.2 Modified Zernike Moments

Zernike moments have been preciously used in the application for other biometrics, such as face and gait recognition, and shown an encouraging performance. Its translation,

rotation and scale invariance promote itself as a widely used feature extraction approach. Zernike moments [15] of order  $p$  with repetition  $q$  of an image intensity function  $f(r,\theta)$  are defined as:

$$Z_{pq} = \frac{p+1}{\pi} \int_0^{2\pi} \int_0^1 V_{pq}(r, \theta) f(r, \theta) r dr d\theta, \quad |r| \leq 1, \tag{4}$$

Where  $r$  is the radius from the image center,  $\theta$  is the angle with the  $x$ -axis in a counter clockwise direction and Zernike polynomials :

$$V_{pq}(r, \theta) = R_{pq}(r) e^{-jq\theta}, \quad j = \sqrt{-1},$$

$$R_{pq}(r) = \sum_{k=0}^{p-|q|} (-1)^k \frac{(p-q)!}{k! \left(\frac{p+|q|}{2} - k\right)! \left(\frac{p-|q|}{2} - k\right)!} r^{p-2k}, \tag{5}$$

where  $0 \leq |q| \leq p$  and  $p - |q|$  is even.

The Zernike moments for a digital image  $f(x,y)$  can be defined as:

$$Z_{pq} = \lambda(p, N) \sum_{x=1}^N \sum_{y=1}^N R_{pq}(r_{xy}) e^{-jq\theta} f(x, y), \quad 0 \leq r_{xy} \leq 1, \tag{6}$$

where  $\lambda(p, N) = \frac{p+1}{N^2}$ ,  $r_{xy} = \sqrt{x^2 + y^2}$ ,  $\theta = \arctan(y/x)$ .

The original Zernike moments are modified (detailed in [16]) as follows:

$$m_{00} = \sum_x \sum_y f(x, y). \tag{7}$$

Normalize the Zernike moments using  $m_{00}$ :

$$Z'_{pq} = \frac{Z_{pq}}{m_{00}}, \tag{8}$$

where  $Z'_{pq}$  are the modified Zernike moments.

The magnitudes of modified Zernike moments  $|Z'_{pq}|$  are regard as the features of the ROI.

### 3.3 Coarse-Level Identification

Identification is a process of comparing one image against  $N$  images. In our system, the identification task is completed in two stages. Given a query sample, it firstly compares with all the templates in database at the coarse-level stage. Then a threshold value is set to select small similar candidates for fine-level identification. The Man-

hattan distance is used to measure the similarity between query sample and the template at coarse-level identification stage, which is defined as:

$$d_M(q, t) = \sum_{i=1}^8 |q_i - t_i|, \quad (9)$$

Where  $q_i$  and  $t_i$  are the  $i$ -th component of the hand shape feature vector of the query sample and the template, respectively. If the distance is smaller than pre-defined threshold value, record the index number of the template into an index vector  $I$  for fine-level identification.

### 3.4 Fine-Level Identification

BPNN is one of the simplest and most general methods for supervised training of multilayer neural networks, which has been widely utilized in the field of pattern recognition. We use this well-known approach to perform the fine-level identification task. The architecture of our proposed network is designed to be a three-layer-based network which includes an input, a hidden and an output layer as depicted in Fig. 5.

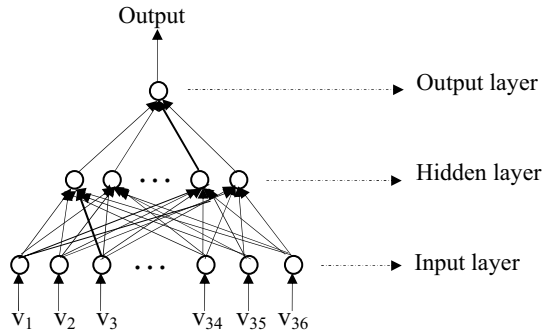


Fig. 5. The architecture of BP neural network

These layers are interconnected by modifiable weights, which are represented by links between layers. In training stage of BPNN,  $M$  image samples of a specific individual  $X$  called positive samples and  $N$  image samples of other  $K$  persons called negative samples are collected to train the BPNN.

In order to reduce the necessary training samples, each neural network corresponds to only one handprint owner. For practical systems, this approach offers another significant advantage: if there is a new person added to database, we only have to train a new BPNN and needn't to retain a very large neural network.

The index vector  $I$  has been recorded in coarse-level identification stage. In this section, the testing image will be further matched with the templates whose index numbers are in  $I$ . We study the first 36 Zernike moments for each ROI. Thus, the 36 Zernike moments of the query sample will input to the BPNN whose index numbers are in  $I$  for fine-level identification.

## 4 Experimental Results

A series of experiments have been done to evaluate the superiority and effectiveness of the proposed approach in a handprint database, which contains 1000 color handprint images collected from 100 individuals' left hand with different strength of brightness. The size of images is  $500 \times 500$  with 100 dip. Five images per user are taken as the database templates and others are used for testing samples.

The performance of our identification system is measured by correct identification rate (*CIR*) and false identification rate (*FIR*). At coarse-level identification stage, a looser threshold value ( $T_1 = 5$  in our experiments) is defined to decide which candidate samples will participate in fine-level identification. For a given testing sample, 83% of the templates in the database are filter out on average. We need to pre-define an output threshold ( $T_2$ ) of BPNN seriously. More than one output of BPNN may larger than  $T_2$ . We select the largest output as the final identification result. If no output is larger than  $T_2$ , it illuminates that the query sample is an attacker. The testing results show that the *CIR* can reach 98.35%. The identification result based on different  $T_2$  at fine-level identification stage is list in Table 1.

**Table 1.** The *CIR* with different  $T_2$  with our proposed methods

$T_2$	<i>CIR</i> (%)
0.75	92.74
0.80	93.19
0.85	96.54
0.92	98.35
0.95	85.62

## 5 Conclusions

In this paper, both hand shape features and palm texture features are used to facilitate a coarse-to-fine dynamic personal identification task. A novel image threshold method is used to segment the hand image from background based on color information. When extracting the texture of ROI, the circular Gabor filter only convolutes with the green component of the color ROI. A set of features which are proved to be rotation and scalar invariant are extracted by the modified Zernike moments. And the normalization is not needed in the preprocessing stage, which can eliminate the errors in both scale and rotation invariance and doesn't destroy the quality of the image. In the fine-level classification stage, one-class-one-network classification structure is implemented for final confirmation. Although the experimental results demonstrated that the proposed approaches are feasible, we will do more experimentation in a bigger handprint database. Novel filters for extracting the texture of palmprint image and feature extraction methods should be proposed for better identification performance. Comparisons between classical methods and our proposed methods are also an important research issue in our future works.

## References

1. X. Wu, D. Zhang, K. Wang, Bo Huang: Palmprint classification using principal lines, *Pattern Recognition* 37 (2004) 1987-1998
2. X. Wu, K. Wang: A Novel Approach of Palm-line Extraction, *Proceedings of International Conference on Image Processing*, 2004
3. Nicolae Duta, Anil K. Jain, Kanti V. Mardia: Matching of palmprint, *Pattern Recognition Letters* 23 (2002) 477-485
4. D. Zhang, W.K. Kong, J. You, M. Wong: On-line palmprint identification, *IEEE Transaction Pattern Analysis and Machine Intelligence* vol. 25 (2003) 1041-1050
5. W. K. Kong, D. Zhang , W. Li: Palmprint feature extraction using 2-D Gabor filters, *Pattern Recognition* 36 (2003) 2339-2347
6. Slobodan Ribaric, Ivan Fratric: A Biometric identification System Based on Eigenpalm and Eigenfinger Features, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005) 1698-1709
7. Xiangqian Wu, David Zhang, Kuanquan Wang: Fisherpalms based palmprint recognition. *Pattern Recognition Letters* 24 (2003) 2829-2838
8. Wenxin Li, Jane You, David Zhang: Texture-Based Palmprint Retrieval Using a Layered Search Scheme for Personal Identification, *IEEE Transactions on Multimedia* 7 (2005) 891-898
9. Jane. You, Wenxin. Li, David. Zhang: Hierarchical palmprint identification via multiple feature extraction, *Pattern Recognition* 35 (2003) 847-859
10. Chao Kan, Mandyam D. Srinath: Invariant character recognition with Zernike and orthogonal Fourier-Mellin moments, *Pattern Recognition* 35 (2002) 143-154
11. W. Li, D. Zhang. Z. Xu: Palmprint identification feature extraction Fourier Transform, *International Journal of Pattern Recognition and Artificial Intelligence* 16 (2003) 417-432
12. Lei, Zhang, David Zhang: Characterization of Palmprints by Wavelet Signatures via Directional Context Modeling, *IEEE Transaction on Systems, Man, and Cybernetics* 34 (2004) 1335-1347
13. C. Sanchez-Avila, R. Sanchez-Reillo: Two different approaches for iris recognition using Gabor filters and multiscale zero-crossing representation, *Pattern Recognition* 38 (2005) 231-240
14. Jainguo Zhang, Tieniu Yan, Li Ma: Invariant Texture Segmentation Via Circular Gabor Filters
15. Ying-Han Pang, Tee Connie, Andrew Teoh Beng Jin, David Ngo Chek Ling: Palmprint authentication with Zernike moment invariants, 199-202
16. N. K. Kamila, S. Mahapatra, S. Nanda: Invariance image analysis using modified Zernike moments, *Pattern Recognition Letters* 26 (2005) 747-753

# White Blood Cell Automatic Counting System Based on Support Vector Machine

Tomasz Markiewicz<sup>1</sup>, Stanisław Osowski<sup>1,2</sup>, and Bożena Mariańska<sup>3</sup>

<sup>1</sup>Warsaw University of Technology,  
Koszykowa 75, 00-662 Warsaw, Poland  
{sto, markiewt}@iem.pw.edu.pl

<sup>2</sup>Military University of Technology, Warsaw, Poland

<sup>3</sup>Institute of Hematology, Warsaw, Poland

**Abstract.** The paper presents the automatic system for white blood cell recognition on the basis of the image of bone marrow smear. The paper proposes the complete system solving all problems, beginning from cell extraction using watershed algorithm, generation of different features based on texture, geometry and the statistical description of image intensity, feature selection using Linear Support Vector Machine and final classification by applying Gaussian kernel Support Vector Machine. The results of numerical experiments of recognition of 10 classes of blood cells of patients suffering from leukaemia have shown that the proposed system is sufficiently accurate so as to find practical application in the hospital practice.

## 1 Introduction

The recognition of the blood cells in the bone marrow of the patients suffering from leukemia is a very important step in the recognition of the development stage of the illness and proper treatment of the patients [5], [16]. The percentage of different cells (so called myelogram) is a major factor in defining various subtypes of leukemias and proper treatment of patients.

There are different cell lines in the bone marrow, from which the most important are the erythrocytic, lymphocytic and granulocytic series. A lot of different blast cell types belonging to these lines have been recognized up to now by the specialists. The most known and recognized cells are: erythroblasts, mono- and myelo-blasts (called shortly blasts), promyelocytes, myelocytes, metamyelocytes, neutrophilic band and segmented, eosinophilic cells, lymphocytes, plasmocyte, etc. They differ in size, texture, shape, density and color.

Up to now no automatic system exists that can recognize and count the blood cells with the accuracy acceptable in hospital practice. Everything is done manually by the human expert and relies on him/her subjective assessment. The acceptable accuracy of the human expert is estimated as approximately 85% (difference of 15% of the results produced by two independent experts). In our paper the results will be referred to the human expert score and the error measures defined with respect to this score, treating it as a base.

Although some attempts to solve the problem of automatic counting of the blood cells have been presented recently [1], [7], [12], [13], [16] but the results are still not

satisfactory and a lot of research needs to be done to achieve the efficiency of the human expert.

The paper presents a fully automatic system of blood cell recognition. The input to the system is the image of the bone marrow of the patient. On the basis of it the individual cells are extracted and then preprocessed to form the diagnostic features used by the neural recognition and classifying network. The final recognizing system is built on the basis of Gaussian kernel Support Vector Machine. The results of recognition of 10 different cell types are given and discussed in the paper. The results confirm good efficiency of the proposed system. At the recognition of 10 types of cells we have achieved the agreement of almost 87% (13% of misclassification) with the human expert score. These results are acceptable in hospital practice.

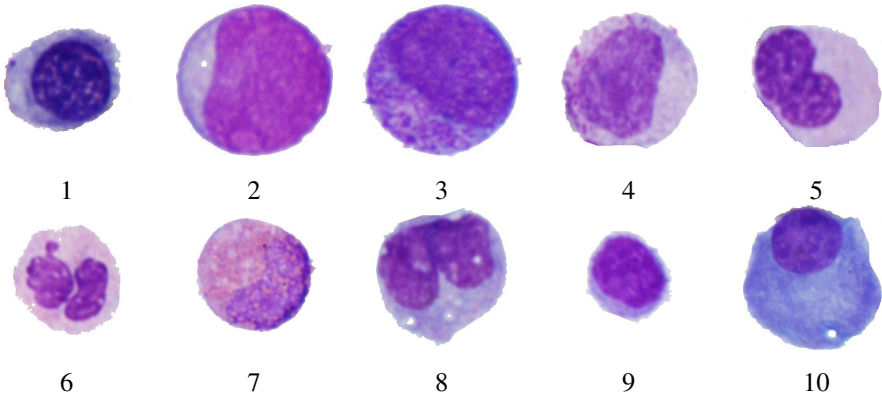
## 2 Preprocessing of the Image for Feature Generation

The first step in automatic recognition of the cells is extraction of the individual blood cells from the bone marrow smear image. We have applied here the image segmentation technique based on the morphological operations arranged in the form of watershed transformation [11]. The applied procedure of the watershed image segmentation and cell separation consists of the following stages [7]:

- Transformation of the original image first to the grey scale and then to binary.
- Application of closing and erosion operations to smooth the contours and to eliminate the distortions.
- Generation of the map of distances from each black pixel to the nearest white pixel.
- Application of the watershed algorithm based on these distances for the final division of the image into catchment's basins, each corresponding to one cell.
- Final extraction of the real cell (in original colour distribution), corresponding to each catchment's basin.

The result of such image preprocessing is the set of individual blood cells existing in the analyzed image. In the numerical experiments we have considered 10 classes of blood cells. They include: erythroblasts (1), mono- and myelo-blasts, called usually blasts (2), promyelocytes (3), myelocytes (4), metamyelocytes (5), neutrophilic band and segmented (6), eosinophilic cells (7), monocytes (8) lymphocytes (9) and plasmocyte (10). The numbers in parentheses denote our notation of the particular class. In further investigations we will refer to the particular class by its number. The considered classes represent different cell lines in bone marrow as well as different stages of development within the same line. To cope with the cells not classified to any of the mentioned above classes, for example the red blood corpuscles, the scarce cells not belonging to any already defined type, the so called shadows of cells deprived of the nucleus, parts of the cut blasts, etc., we have created the heterogeneous class denoted by 11. Fig. 1 presents the exemplary cells belonging to 10 considered classes. Looking at the whole family of cells we may notice, that the main differences among the blood cells are concerned with the shape, size, granules, texture, color and intensity of the images associated with different cell types.





**Fig. 1.** The exemplary blood cell representatives used in experiments and their numeric notation

However the great problems of recognition appear for the cells belonging to the same class. The parameters of cells belonging to the same class vary a lot. Table 1 presents the mean value (upper number) and standard deviation (lower number) calculated for over 6000 samples of chosen parameters for all considered classes.

**Table 1.** The mean values and standard deviations of the chosen parameters of the cells forming the data base

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Class 10
$f_1$	0.043 $\pm 0.037$	0.116 $\pm 0.041$	0.184 $\pm 0.060$	0.115 $\pm 0.042$	0.079 $\pm 0.032$	0.056 $\pm 0.021$	0.144 $\pm 0.053$	0.108 $\pm 0.022$	0.056 $\pm 0.015$	0.128 $\pm 0.052$
$f_2$	0.111 $\pm 0.072$	0.021 $\pm 0.017$	0.029 $\pm 0.063$	0.065 $\pm 0.031$	0.083 $\pm 0.036$	0.136 $\pm 0.055$	0.021 $\pm 0.026$	0.086 $\pm 0.033$	0.034 $\pm 0.021$	0.057 $\pm 0.057$
$f_3$	0.007 $\pm 0.007$	0.020 $\pm 0.014$	0.047 $\pm 0.033$	0.042 $\pm 0.027$	0.037 $\pm 0.027$	0.035 $\pm 0.018$	0.028 $\pm 0.019$	0.031 $\pm 0.014$	0.010 $\pm 0.007$	0.031 $\pm 0.021$
$f_4$	0.009 $\pm 0.012$	0.011 $\pm 0.014$	0.039 $\pm 0.035$	0.041 $\pm 0.035$	0.025 $\pm 0.024$	0.031 $\pm 0.023$	0.029 $\pm 0.039$	0.017 $\pm 0.015$	0.006 $\pm 0.007$	0.032 $\pm 0.031$
$f_5$	0.319 $\pm 0.147$	0.603 $\pm 0.117$	0.598 $\pm 0.105$	0.605 $\pm 0.096$	0.551 $\pm 0.092$	0.502 $\pm 0.104$	0.603 $\pm 0.105$	0.564 $\pm 0.080$	0.489 $\pm 0.123$	0.539 $\pm 0.109$
$f_6$	0.281 $\pm 0.115$	0.117 $\pm 0.063$	0.205 $\pm 0.118$	0.187 $\pm 0.107$	0.231 $\pm 0.116$	0.263 $\pm 0.106$	0.244 $\pm 0.119$	0.143 $\pm 0.065$	0.146 $\pm 0.078$	0.140 $\pm 0.078$
$f_7$	0.354 $\pm 0.144$	0.556 $\pm 0.087$	0.447 $\pm 0.102$	0.358 $\pm 0.073$	0.352 $\pm 0.077$	0.471 $\pm 0.176$	0.391 $\pm 0.093$	0.413 $\pm 0.063$	0.450 $\pm 0.099$	0.379 $\pm 0.109$

These parameters include: the area of nucleus ( $f_1$ ), the ratio of the area of cell and the area of nucleus ( $f_2$ ), symmetry of the nucleus ( $f_3$ ), compactness of nucleus ( $f_4$ ), mean ( $f_5$ ), variance ( $f_6$ ) and skewness ( $f_7$ ) of the histogram of the red color of the cell. Each row represents the chosen parameter and column – the appropriate class. Some parameters (for example  $f_3$ ,  $f_4$ ,  $f_6$ ) have large standard deviations with respect to their mean values. It means large dispersion and some difficulties in associating them with the proper class.

The next step of cell image preprocessing is generation of the diagnostic features well characterizing different cells. In our approach to feature characterization we will rely on the features belonging to three main groups, of the textural, geometrical and statistical nature.

The textural features refer to an arrangement of the basic constituents of the material and in the digital image are depicted by the interrelationships between spatial arrangements of the image pixels [15]. They are seen as the changes in intensity patterns or the gray tones. There are many different methods of texture description. In our approach we have chosen the Unser features and Markov random field descriptors [7].

The next set of parameters corresponds to the geometrical shape of the nucleus and of the whole cell. For this characterization of the blood cells we have used the following parameters: radius, perimeter, the ratio of the perimeter and radius, area, the area of convex part of the nucleus, filled area, compactness, concavity, number of concavity points, symmetry, major and minor axis lengths.

The last set of features has been generated from the analysis of the intensity distribution of the image. The histograms and gradient matrices of such intensity have been determined for three color components R, G and B of the image. On the basis of such analysis the following features have been generated: mean value, variance, skewness and kurtosis of histogram of the image and histogram of the gradient matrix of the image. Applying these techniques of feature generation we have got more than 150 different features.

The excessive numbers of features, some of which may be contradictory and represent the noise or insignificant information, is harmful for the recognition process of the patterns [2]. Thus the main problem in classification and machine learning is to reduce this number to the optimal one appropriate to the problem. Note that elimination of less significant features leads to the reduction of the dimensionality of the feature space, improvement of generalization ability and better performance of the classifier in the testing mode on the data not taking part in learning. Among many different feature selection methods [3], [10] the best results have been obtained at application of the linear Support Vector Machine. In this approach the ranking of the features is done considering simultaneously the whole set of candidate features generated in the introductory phase. They are applied simultaneously for learning linear SVM. After learning the weights are sorted according to their absolute values. The feature ranking is based on the idea that the feature associated with the larger weight is more important than that associated with the smaller one.

The features selected in the above stage of experiments have been used as the input signals to the Gaussian kernel Support Vector Machine [9,14] used as the classifier. For recognition of 11 classes we have applied the one-against-one approach [4], in which we train many local two-class recognition classifiers, on the basis of which the final winner is selected. At  $M$  classes we have to train  $M(M-1)/2$  two-class SVM based recognizing networks. The majority vote across all trained classifiers is applied to find the final winner at the presentation of the input vector  $\mathbf{x}$ . The training of the SVM networks has been provided by Hsu-Lin BSVM algorithm [4].

### 3 The Results of the Numerical Experiments

The numerical experiments have been performed on the data of the blood cells created in cooperation with the Institute of Hematology in Warsaw. The bone marrow smear samples have been collected from 54 patients suffering from AML-M2, AML-M3, AML-M4, AML-M5, CLL, ALL, plasmocytoma and lymphoma. The images of the bone marrow were digitized using an Olympus microscope with the magnification of 1000x and a digital camera of resolution 1712x1368 pixels. The picture was saved in RGB format. The smears were processed by applying the standard May-Grunwald-Giemsa (MGG) method. The experiments were performed in two stages. The data of the first set of 40 patients were used in the introductory learning and testing trials directed on a versatile verification of the developed system. After designing the recognizing system the next set of 14 patients was used to produce data for testing the developed automatic system. Table 2 shows the number of cells available for different classes in the first part of experiments. The number of samples belonging to different classes changes a lot from class to class. Some of them were very sparsely represented in the bone marrow of patients, mainly due to their rare occurrence in the typical process of development of the illness.

**Table 2.** The number of samples of each class of cells used in the first set of experiments

Class	1	2	3	4	5	6	7	8	9	10	11
Number of samples	754	114	92	151	125	365	121	281	93	140	202

The first set of experiments, relying on training and validation of the Gaussian kernel SVM, was aimed on the assessment of the classification accuracy on the closed set of data depicted in Table 2. The whole data set has been split into 5 groups containing approximately the same number of representatives of individual classes and the cross validation strategy has been applied in experiments. Four groups have been combined together and used in the learning, while the fifth one used only in testing. The experiments have been repeated five times exchanging the contents of the 4 learning subsets and the testing subset. The optimal values of hyper parameters of SVM (the constant  $\gamma$  of the Gaussian function, the regularization constant C) have been adjusted in the introductory stage of the learning procedure using certain part of learning data (validation data). The misclassification ratio in learning and testing modes has been calculated as the mean of all 5 runs.

In the experiments we have tried different numbers of the most important features. The most promising results have been obtained by applying only 30 of the best features, ranked by the linear SVM network. The total percentage error on the learning data has reached the value of 9.9% and the testing data of 13.6%. The detailed results of the recognition of individual classes of cells after application of this type of feature ranking are presented in Table 3.

Some blood types have been recognized with large errors (for example classes No 3 and 5). There are two sources of errors. The worst recognized classes are usually those for which small number of representatives is available in the learning set, and to

such case belongs class No 3 and 5. The second reason follows from the similarities among the neighbouring blood cells in their development line. The transition from one cell type to the neighbouring one is continuous and even experts have difficulties distinguishing between them.

**Table 3.** The detailed results of the blood cell recognition in the cross validation approach

Class	1	2	3	4	5	6	7	8	9	10	11
Number of testing errors	16	15	29	37	51	26	4	14	10	15	6
Percentage rate [%]	2.1	13.1	31.5	24.5	40.8	7.1	3.3	4.9	10.7	4.3	7.4

In our data set there are cells close to each other in their development stage. For example the cells belonging to classes 2, 3, 4, 5 and 6 represent the succeeding stages of cells within the granulocytic development line. Recognition between two neighbouring cells in the same development line is dubious even for the human expert, since the images of both cells are very alike and thus difficult to recognize. Close analysis of our misclassification cases reveals that most errors have been committed at the recognition of the neighbouring cells in their development lines.

**Table 4.** The sample confusion matrix of the cell recognition for the testing data

Cl <sub>a</sub>											
1	738	2	1			3			2	5	3
2	1	99	4	3	1			3		1	2
3	2	6	63	17	1					2	1
4	2	2	6	114	14	7			2	2	2
5			1	24	74	24					2
6			1	5	18	339		1			1
7				1		1	117				2
8	2	2		2				267			8
9	2	3		5					83		
10	3			2						134	1
11	1			6	1	1		6			187

Table 4 presents the details of the class recognition related to the optimal set of features following from the SVM ranking. It is done in the form of so called confusion matrix, represented as the summed results of the cross validation experiments for the testing data. The rows represent the real classes, while columns – outputs of the classifier. The diagonal entries represent the numbers of properly recognized classes. Each entry outside diagonal represents error. The entry in the (i,j)th position of the matrix means false assignment of ith class to the jth one. As we see most errors have been committed for the neighboring classes (the data in the region encircled by the bold closed line in the table 4).

Neglecting the errors committed between the neighboring classes we got the significant reduction of errors for each class and for the whole set. Table 5 presents the composition of the testing errors for this case. The average mean error for all classes has been reduced from 13.6% to 5.9% only.

**Table 5.** The summary of performance of the classifier at the testing data for 11 classes neglecting the errors between neighboring cells in their development line

Class	1	2	3	4	5	6	7	8	9	10	11
Number of testing errors	16	11	6	17	3	8	4	14	10	15	6
Percentage rate [%]	2.1	9.6	6.5	11.3	2.4	2.2	3.3	4.9	10.7	4.3	7.4

The next set of experiments has been performed on the established and trained Gaussian kernel SVM system for the new data not taking part in previous experiments. The data has been acquired from the new 14 patients of the hospital. In each case different number of cells has been acquired. Some of the cell families were absent or very scarce (just few cells) for some patients. For all new patients the automatic analysis of the blood cells has been performed using our system and the results have been compared with the results of human expert.

Table 6 presents the summary of results for all 14 patients. Column 3 presents the average discrepancy ratio ( $\epsilon_1$ ) counted for all cells and column 4 – the discrepancy rate ( $\epsilon_2$ ) neglecting the errors committed for the neighboring cells. The mean values of  $\epsilon_1$  and  $\epsilon_2$  for all 14 patients are equal 18.14% and 14.78%, respectively.

**Table 6.** The summary of results for 14 patients

Patient	Number of cells	$\epsilon_1$	$\epsilon_2$
1	188	13.30%	11.70%
2	223	16.59%	12.56%
3	200	29.50%	29.00%
4	195	13.33%	10.77%
5	310	8.71%	6.13%
6	152	23.68%	13.16%
7	376	27.13%	25.53%
8	186	23.12%	20.43%
9	200	14.00%	9.50%
10	278	28.78%	17.99%
11	167	23.55%	22.75%
12	163	10.43%	10.43%
13	260	11.72%	9.62%
14	268	10.45%	8.58%
Total	3166	<b>18.14%</b>	<b>14.78%</b>

Table 7 presents the detailed results (the mean discrepancy rates) concerning the recognition of all individual classes of cells, averaged over all patients.

**Table 7.** The mean discrepancy rate of 10 cell families for all patients

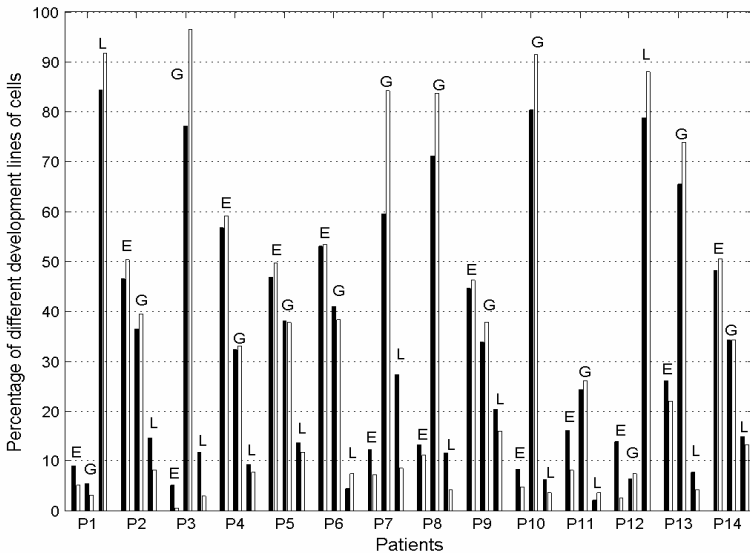
Cell	1	2	3	4	5	6	7	8	9	10
$\epsilon_1$ [%]	4.76	29.40	40.63	48.30	46.88	8.24	19.05	23.66	16.07	3.91
$\epsilon_2$ [%]	4.76	27.30	28.13	22.16	14.06	1.76	19.05	23.66	16.07	3.91

Table 8 presents the detailed results of the composition of myelograms for 5 patients obtained by the human expert (HE) and our automatic system (AS). As we see in most cases the results are well compatible, although we don't know which of them represents the true number.

**Table 8.** The detailed myelograms of 5 patients (AS – automatic system, HE – human expert)

Cell	Patient 1		Patient 2		Patient 3		Patient 4		Patient 5	
	AS	HE	AS	HE	AS	HE	AS	HE	AS	HE
1	15	9	78	79	72	68	40	32	70	72
2	1	0	4	4	1	0	58	60	2	5
3	1	3	12	6	7	6	1	4	5	5
4	4	0	9	20	9	9	9	8	6	9
5	2	0	14	13	9	13	2	4	7	7
6	0	1	14	13	13	10	24	24	30	28
7	1	1	8	6	2	0	6	4	2	5
8	2	0	4	3	2	0	1	0	2	0
9	18	19	19	13	12	9	11	7	26	22
10	120	123	4	1	0	0	1	0	6	4

Figure 2 presents the cumulative myelograms for all 14 patients obtained by our automatic system (black bars) and human expert (transparent bars). The letter: L, G, E denote the lymphocytic, granulocytic and erythrocytic development lines of the blood, respectively, and P1, P2, ..., P14 – the notations of patients.



**Fig. 2.** The myelograms of 14 patients generated by the system and human expert

As it is seen the cumulative myelograms (important in medical practice) are well compared and satisfy the tolerance requirement of approximately 15%.

## 4 Conclusions

The paper has presented the automatic system of the blood cell recognition on the basis of the image of the bone marrow. It applies the morphological preprocessing of the image for individual cell extraction, generation and selection of the diagnostic features and the recognition system using Gaussian kernel SVM working in one against one mode. The results of numerical experiments for recognition of 11 classes of blood cells have shown that the system works well, delivering the satisfactory results. The final accuracy of the proposed automatic recognition system is now comparable to the accuracy of the human expert. Also application of the system to the myelogram preparation of newly acquired data has shown good agreement with the human expert score.

## References

1. Beksac M., Beksac M. S., Tippi V. B., Duru H. A., Karakas M. U., Nurcakar, A.: An Artificial Intelligent Diagnostic System on Differential Recognition of Hematopoietic Cells from Microscopic Images. *Cytometry*, 30 (1997), 145-150
2. Duda R. O., Hart P. E., Stork P.: *Pattern Classification and Scene Analysis*. Wiley N.Y. (2003)
3. Guyon I., Weston J., Barnhill S., Vapnik V.: Gene Selection for Cancer Classification Using Support Vector Machines. *Machine Learning*, 46 (2002), 389-422
4. Hsu C. W., Lin C. J.: A Comparison Methods for Multi Class Support Vector Machines. *IEEE Trans. Neural Networks*, 13 (2002), 415-425
5. Lewandowski K., Hellmann A.: *Haematology Atlas*, Multimedia Medical Publisher. Gdansk (2001)
6. Matlab user manual – Image processing toolbox, MathWorks, Natick (1999)
7. Markiewicz T., Osowski S., Mariańska B., Moszczyński L.: Automatic Recognition of the Blood Cells of Myelogenous Leukemia Using SVM. *IJCNN Montreal*, (2005), 2496-2501
8. Osowski S., Markiewicz: Support Vector Machine for Recognition of White Blood Cell of Leukemia (in Camps-Valls G., Rojo-Alvarez L., Martinez-Ramon M. (Eds.), *Kernel Methods in Bioengineering, Communication and Image Processing*), Idea Group (2007), 81-108
9. Schölkopf B., Smola A.: *Learning with Kernels*, MIT Press, Cambridge, MA (2002)
10. Schurmann J.: *Pattern Classification, a Unified View of Statistical and Neural Approaches*. Wiley N. Y. (1996)
11. Soille P.: *Morphological Image Analysis, Principles and Applications*. Springer Berlin (2003)
12. Theera-Umpon N., Gader P.: System-level Training of Neural Networks for Counting White Blood Cells. *IEEE Trans. SMS-C*, 32 (2002), 48-53
13. Ushizima D., M., Lorena A. C., de Carvalho A. C. P.: Support Vector Machines Applied to White Blood Recognition. *V Int. Conf. Hybrid Intelligent Systems*, Rio de Janeiro (2005), 234-237
14. Vapnik V.: *Statistical Learning Theory*. Wiley N.Y. (1998)
15. Wagner T.: *Texture Analysis* (in Jahne B., Haussecker H., and Geisser P., (Eds.), *Handbook of Computer Vision and Application*), Academic Press (1999), 275-309
16. DiffMaster Octavia DM96, Cellavision AB, [www.cellavision.com](http://www.cellavision.com)

# Kernels for Chemical Compounds in Biological Screening

Karol Kozak<sup>1</sup>, Marta Kozak<sup>1</sup>, and Katarzyna Stapor<sup>2</sup>

<sup>1</sup> Max Planck Institute of Molecular Cell Biology,  
01307 Dresden, Germany  
{kozak, mkozak}@mpi-cbg.de  
<http://www.mpi-cbg.de>

<sup>2</sup> Silesian Technical University,  
44-100 Gliwice, Poland  
katarzyna.stapor@polsl.pl

**Abstract.** Kernel methods are a class of algorithms for pattern analysis with a number of convenient features. This paper proposes extension of the kernel method for biological screening data including chemical compounds. Our investigation of extending kernel aims to combine properties of graphical structure and molecule descriptors. The use of such kernels allows comparison of compounds, not only on graphs but also on important molecular descriptors. Our experimental evaluation of eight different classification problems shows that a proposed special kernel, which takes into account chemical molecule structure and molecule descriptors, statistically improves significantly the classification performance.

## 1 Introduction

Biological screening processes are used in drug discovery to screen large numbers of compounds against a biological target. By quantifying structure-activity relationship (QSAR) means a group of bioinformatics or chemoinformatics algorithms, typically using physico-chemical parameters or three-dimensional molecular fields with statistical techniques. Finding active compounds amongst those screened, QSAR algorithms are aimed at planning additional active compounds without having to screen each of them individually. Classically, QSAR models are designed by representing molecules by a large set of descriptors, i.e. by a high dimensional vector, and then applying some kind of Pattern Recognition methods like Neural Networks, Decision Trees or, more recently, Support Vector Machines and K-Nearest Neighbors. Modeling the relationship between the chemical structure, descriptors and activities of sampled compounds provides an insight into the QSAR, gives predictions for untested compounds of interest, and guides future screening.

The first step in the process of determining features of chemical compounds which are important for biological activity is to describe the molecules in a relevant, quantitative manner. To get numbers from molecule we need molecular structure and descriptors that can effectively characterize the molecular size, molecular branching or the variations in molecular shapes, and can influence the structure and its activities. By chemical structure we mean all characteristics that define a molecule, including atoms in the molecule, bonds between atoms, the three-dimensional configuration of the atoms and so forth.



Clearly the structures present in screening data are essential for the more-or-less automated extraction of meaning, patterns, and regularities using machine learning methods. Here we focus on graph-structured data on screening compound and the development and application of kernel methods for graph-structured data in combination with molecular descriptors, with particular emphasis on QSAR in the screening process and the prediction of biological activity of chemical compounds.

Because chemical compounds are often represented by the graph of their covalent bonds, pattern recognition methods in this domain must be capable of processing graphical structures with variable size. One way of doing so is using a symmetric, positive kernel e.g. Schoelkopf & Smola, 2002 [21]. Additionally Froehlich et al. [8] propose a kernel function for chemical compounds between labeled graphs, which they call Optimal Assignment (OA) kernel. Selecting optimal similarity features of a molecule based on molecular graph or descriptors is a critical and important step, especially if one is interested in QSAR studies. It has been shown [1, 6, 11, 16] that the quality of the inferred model strongly depends on the selected molecular properties (graph or descriptors). Nevertheless, after our examination of different molecules used in screening experiments, where the structure of the molecule is very complex, grouping of atoms, bonds proposed, based on OA kernel is not always relevant. In such cases we needed additional criteria to extend OA kernel also including molecule descriptors. As far as we know there is no extension of OA kernel where we should also consider molecule descriptors. Descriptors play an important role in the prediction of activity in screening data like constitutional descriptors, RDF descriptors and topological descriptors [25]. One of the newest additions to this class of whole-molecule descriptors is the set of Burden metrics of Pearlman and Smith 1998 [18]. Pearlman and Smith showed that it may be possible to find fairly low-dimensional (2-D or 3-D) subsets of Burden variables [3] such that active compounds are clustered in the relevant subspace. Though a relatively "coarse" description of a molecule, the Burden numbers are attractive because of their one-dimensional nature and the comparative ease of their computation. Moreover, two molecules with close Burden numbers often appear similar when comparing their chemical structures (e.g., by comparing numbers of fragments or functional groups two molecules have, or have not, in common). The goal of our work is to define a kernel for chemical compounds, which like the OA graph kernel, is of general use for QSAR problems, but is more focused on screening data. In developing a good quality QSAR model for these data we combined OA graph kernel and kernel where we considered Burden descriptors.

This paper is organized as follows: First, we briefly review the literature on the study of kernels. Next we review the basic theory of kernel functions. In Sect. 4 we explain our extensions to the basic approach. In Sect. 5 we experimentally evaluate our approach and compare it to classical models with graph kernel and models based on descriptors. The conclusion is given will in Sect. 5.

## 2 Related Work

Finally, in recent years, kernel methods have emerged as an important class of machine learning methods suitable for variable-size structured data [12, 22], (Schoelkopf and Smola, 2002) [21]. Much of the existing work on graph kernels is

concentrated on the special cases where the graph is a string, e.g. (Joachims, 2002) [24], or the graph is a tree, e.g. Vishwanathan & Smola, 2003 [20]. Surprisingly little work has been done on graphical structures. Most of the work to date on general graph kernels has been done by Gaertner (2003) [23] using the theory of positive definite kernels and kernel methods. Author introduced positive definite kernels between labeled graphs, based on the detection of common paths between different graphs. These kernels correspond to a dot product between the graphs mapped to an infinite-dimensional feature space, but can be computed in polynomial time with respect to the graph sizes. In this study we investigated the work of Froehlich et al. (2005) [8] because it focuses on graph kernel for molecular structure and straightforward to implement.

### 3 Kernel Methods

Given two input objects  $m$  and  $m'$ , such as two molecules, the basic idea behind kernel methods is to construct a kernel  $k(m, m')$  which measures the similarity between  $m$  and  $m'$ . This kernel can also be viewed as an inner product of the form  $k(m, m') = (\Phi(m), \Phi(m'))$  in an embedding feature space determined by the map  $\Phi$  which needs not be given explicitly. Regression, classification, and other tasks can then be tackled using linear (convex) methods based solely on inner products computed via the kernel in the embedding space, rather than the original input space. The challenge for kernel methods is to build or learn suitable kernels for a given task. Applications of kernel methods to graphical objects, such as molecular bond graphs require the construction of graph kernels which functions that are capable of measuring similarity between graphs with labeled nodes and edges.

### 4 Graph Kernel

The graphical model approach is a probabilistic approach where random variables are associated with the nodes of a graph and where the connectivity of the graph is directly related to Markovian independence assumptions between the variables.

Here we introduce a graph kernel presented by Mahe et al. (2003) [5]. A labeled graph  $G = (V, E)$  is defined by a finite set of vertices  $V$  (of size  $|V|$ ), a set of edges  $E \subset V \times V$ , and a labeling function  $l : V \cup E \rightarrow A$  which assigns a label  $l(x)$  to any vertex or edge  $x$ . We assume below that a set of labels  $A$  has been fixed, and consider different labeled graphs. For a given graph  $G = (V, E)$ , we denote by  $d(v)$  the number of edges emanating from the vertex  $v$  (i.e., the number of edges of the form  $(v, u)$ ), and by  $V^* = \bigcup_{n=1}^{\infty} V^n$  the set of finite-length sequences of vertices. A path  $h \in V^*$  is a finite-length sequence of vertices  $h = v_1 \dots v_n$  with the property that  $(v_i, v_{i+1}) \in E$  for  $i = 1 \dots n-1$ . We note  $|h|$  the length of the path  $h$ , and  $H(G) \in V^*$  the set of all paths of a graph  $G$ . The labeling function  $l : H(G) \rightarrow A$  can be extended as a function  $l : H(G) \rightarrow A^*$  where the label  $l(h)$  of a path  $h = v_1 \dots v_n \in H(G)$  is the succession of labels of the vertices and edges of the path:  $l(h) = (l(v_1), l(v_1; v_2), l(v_2), \dots, l(v_{n-1}, v_n), l(v_n)) \in A^{2n-1}$ . A positive definite kernel on a space  $X$  is a symmetric function  $K : X^2 \rightarrow \mathbb{R}$  that satisfies  $\sum_{i,j=1}^n a_i a_j K(x_i, x_j)$  for any choice of  $n$  points  $x_1, \dots, x_n \in X$  and coefficients  $a_1, \dots, a_n \in \mathbb{R}$ .

## 4.1 Graph Kernel for Chemical Compounds

Graph kernel is usually made of labeled vertices or nodes and labeled edges (connect nodes). For chemical compounds it is, respectively, atom/nodes labels and bond/edge labels. More naturally, the topology of chemical compounds can be represented as labeled graphs, where edge labels correspond to bond properties like bond order, length of a bond, and node labels to atom properties, like partial charge, membership to a ring, and so on. This representation opens up the opportunity to use graph mining methods (Washio & Motoda, 2003) [27] to deal with molecular structures.

In this section we define the basic notations and briefly review the molecule graph kernel introduced by Froehlich et al. (2003) [8]. The goal is to define a kernel for chemical compounds, which, like the marginalized graph kernel [5], is of general use for QSAR problems, but better reflects a chemists' point of view on the similarity of molecules (Froehlich et al. 2005) [8].

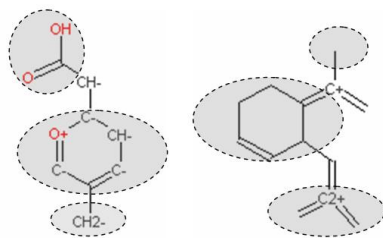


Fig. 1. Matching regions of two molecular structures

Let  $G$  be some domain of structured graph objects. Let us denote the parts of some object  $g$  (e.g. the nodes of a graph) by  $g_1, \dots, g_{|g|}$ , i.e.,  $g$  consists of  $|g|$  parts, while another object  $u$  consists of  $|u|$  parts. Let  $G'$  denote the domain of all parts, i.e.,  $g_i \in G'$  for  $1 < i < |g|$ . Further let  $\pi$  be some permutation of either a  $|g|$  subset of natural numbers  $\{1; \dots; |u|\}$  or a  $|u|$  - subset of  $\{1; \dots; |g|\}$ . Let  $k_1 : G' \times G' \rightarrow R$  be some non-negative, symmetric and positive semidefinite kernel. Then  $k_A : G \times G \rightarrow R$  with

$$k_A(g, u) := \begin{cases} \max_{\pi} \sum_{i=1}^{|g|} k_1(g_i, u_{\pi(i)}) & \text{if } |u| \geq |g| \\ \max_{\pi} \sum_{j=1}^{|u|} k_1(g_{\pi(j)}, u_j) & \text{otherwise} \end{cases} \quad (1)$$

This kernel tries to reach maximal weighted bipartite matching (optimal assignment) of the parts of two objects. Each part of the smaller of both structures is assigned to exactly one part of the other structure, such that the overall similarity score.

## 4.2 Optimal Assignment (OA) Kernels for Chemical Molecules

Let us assume now we have two molecules  $m$  and  $m'$ , which have atoms  $a_1, \dots, a_{|m|}$  and  $a'_1, \dots, a'_{|m'|}$ . Let us further assume we have a kernel  $k_{nei}$ , which compares a pair of atoms  $(a_h, a'_h)$  from both molecules, including information on their neighborhoods,

membership to certain structural elements and other characteristics. Then a valid kernel between  $m$ ,  $m'$  is the optimal assignment kernel:

$$k_{OA}(m, m') := \begin{cases} \max_{\pi} \sum_{h=1}^{|m|} k_{nei}(a_h, a'_{\pi(h)}) & \text{if } |m'| \geq |m| \\ \max_{\pi} \sum_{j=1}^{|m'|} k_{nei}(a_{\pi(h)}, a'_{h'}) & \text{otherwise} \end{cases} \quad (2)$$

where  $k_{nei}$  is calculated based on two kernels  $k_{atom}$  and  $k_{bond}$  which compare the atom and bond features, respectively. Each atom of the smaller of both molecules is assigned to exactly one atom of the larger molecule such that the overall similarity score is maximized. To prevent larger molecules automatically achieving a higher kernel value than smaller ones, kernel is normalized (Schoelkopf & Smola, 2002) [21], i.e.

$$k_{OA}(m, m') \leftarrow \frac{k_{OA}(m, m')}{\sqrt{k_{OA}(m, m)k_{OA}(m', m')}} \quad (3)$$

After the examination of different molecules used in screening experiments, where the structure of the molecule is very complex, grouping of atoms, bonds proposed in OA is not always relevant. In such cases we extended OA kernel with additional criteria which is more relevant for screening data.

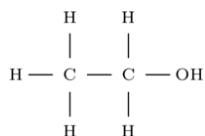
### 4.3 Extension of OA Kernel

A drug-like molecule is a small three-dimensional object that is often drawn as a two-dimensional structure. This two dimensional graph is subject to mathematical analysis and can give rise to numerical descriptors to characterize the molecule. The relationship between descriptors and activity is extremely complex for screening data, and there are several challenges in statistical modeling. First, the potent compounds of different chemical classes may be acting in different ways. Different mechanisms might require different sets of descriptors within particular regions (of the descriptor space) to operate, and a single mathematical model is unlikely to work well for all mechanisms. [25]. Also, activity may be high for only very localized regions. Second, even though a design or screen may include thousands of compounds, it will usually have relatively few active compounds. The scarcity of active compounds makes identifying these small regions difficult. Third, there are many descriptors and they are often highly correlated. This is the case for Burden numbers. Ideally, the descriptors will contain relevant information and be few in number so that the subsequent analysis will not be too complex. For our model we use Burden descriptors based on the work of Burden (1989) [3] to describe the compounds. Burden numbers are very important descriptors for screening data and they are often highly correlated. The Burden descriptors are eigenvalues from connectivity matrices derived from the molecular graph. The square connectivity matrix for a compound has a diagonal element for each heavy (non-hydrogen) atom. The diagonal values are atomic properties, such as size, atomic number, charge, etc. Off diagonal elements measure the degree of connectivity between two heavy atoms. Since eigenvalues are matrix invariants, these numbers measure properties of the molecular graph and hence

the molecule. Burden numbers are continuous variables that describe the features of the compounds such as their surface areas, bonding patterns, charges, and hydrogen bond donor and acceptor ability. Though a relatively “coarse” description of a molecule, the Burden numbers are attractive because of their one-dimensional nature and the comparative ease of their computation. Moreover, two molecules with close Burden numbers often appear similar when comparing their chemical structures (e.g., by comparing numbers of fragments or functional groups two molecules have and have not in common). Pearlman and Smith (1998) [18] showed that it may be possible to find fairly low-dimensional (2-D or 3-D) subsets of Burden variables such that active compounds are clustered in the relevant subspace.

The method for quantitative description of chemical compounds, useful for relating characteristics of molecules to their biological activity was introduced in [3]. Here we briefly summarize these ideas. For every chemical molecule we define its connectivity matrix  $[B]$  composed using of the following rules:

1. Hydrogen atoms are not included
2. Atomic numbers of heavy atoms, arbitrarily indexed, appear on the diagonal of  $[B]$
3. Elements of  $[B]$  corresponding to atom connections are 0.1 for single bonds, 0.2 for double bonds, 0.3 for triple bonds and 0.15 for aromatic delocalized aromatic bonds
4. Elements  $[B]$  corresponding to terminal bonds are augmented by 0.01
5. All other elements are set to 0.001



**Fig. 2.** Ethyl alcohol molecule

As an example, for the molecule of ethyl alcohol shown in Fig. 2 the corresponding connectivity matrix is as follow

$$[B] = \begin{bmatrix} 12 & 0.11 & 0.001 \\ 0.11 & 12 & 0.11 \\ 0.001 & 0.11 & 16 \end{bmatrix}. \quad (4)$$

After the defining connectivity matrix as above, one solves the following eigenvalue problem

$$[B]V_i = V_i\lambda_i \quad (5)$$

where  $\lambda_i$  is  $i$ -th eigenvalue and  $V_i$  is  $i$ -th eigenvector of  $[B]$ . By definition,  $n$  lowest eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  form the set of  $n$  Burden descriptors. So for our example (4) the set of 2-Burden descriptors is [11.8885 12.1084]. Generally we have generated eight chemical descriptor variables using the joelib cheminformatic library [28].

Now we will describe how to compute the kernel efficiently using Burden descriptors. Again, given two molecules  $m$  and  $m'$ , the basic idea of our method is to construct a kernel  $k(m, m')$  which measures the similarity between  $m$  and  $m'$ . This kernel can also be viewed as an inner product of the form  $k(m, m') = (\Phi(m), \Phi(m'))$  in an embedding feature space of Burden descriptors. A kernel can therefore be thought of as a measure of similarity between two molecules: the larger are closer in the feature space. This is a function  $K(\cdot)$  of the distances  $d$  (Euclidean) with maximum in  $d = 0$  and values, which get smaller with growing absolute value of  $d$ .

Our experience has indicated that if the uniform kernel is used, many observations using Burden descriptors can be tied together in terms of their rank score; this does not produce an effective classification. Significant improvements can, therefore, be obtained by choosing Gaussian or triangular Kernel. We take the Gaussian kernel as an example to formulate the kernel computation and then generalize the formulation to all other kernels. This kernel was chosen because it readily produces a closed decision boundary, which is consistent with the method used to select the molecular descriptors. In addition, there is usually no significant difference between the Gaussian and the triangular kernel, making the Gaussian kernel the most attractive choice due to its compact local support. Typical examples for this kind of function  $K(d)_g$  ( $g = 1, \dots, 8$ ) using Gaussian kernel is calculated by:

$$k_{Gaus}(m, m') = \exp(-\gamma |m - m'|) \quad (6)$$

Calculating the OA graph kernel and at same time the Gaussian kernel with eight Burden features give us two ways of comparing molecules: looking at the chemical structure and looking for molecule activity. Having two values from these two kernels and making, at the same time an average, gives us more precise molecule similarity information.

$$k_{scr}(m, m') = \frac{k_{Gaus}(m, m') + k_{OA}(m, m')}{2} \quad (7)$$

## 5 Experimental Evaluation

We experimentally evaluated the performance of kernels in a classification algorithm and compared it against that achieved by earlier approaches on a variety of chemical compound datasets.

### 5.1 Datasets

We used two different public available datasets to derive a total of eight different classification problems. The first dataset was obtained from the National Cancer Institute's DTP AIDS Anti-viral Screen program [4, 19]. Each compound in the dataset is evaluated for evidence of anti-HIV activity.

The second dataset was obtained from the Center of Computational Drug Discovery's anthrax project at the University of Oxford [7]. The goal of this project was to discover small molecules that would bind with the heptameric protective antigen component of the anthrax toxin, and prevent it from spreading its toxic

effects. For these datasets we generated 8 features, called Burden descriptors [3] (eight dimensional spaces).

## 5.2 Results

We tested a proposed extension of molecule graph kernel in two benchmark experiments of chemical compound classification. We use the standard SVM algorithm for binary classification described previously. The regularization factor of SVM was fixed to  $C = 10$ . In order to see the effect of generalization performance on the size of training data set and model complexity, experiments were carried out by varying the number of training samples (100, 200, 400) according to a 5-fold cross validation evaluation of the generalization error.

**Table 1.** Recognition performance on Screening Graph Kernel ( $K_{scr}$ ) by varying the number of training samples (100, 200, 400). Numbers represent correct classification rate [%].

$\gamma$	100		200		400	
	DTP	Toxic	DTP	Toxic	DTP	Toxic
0.1	78.8	73.8	79.8	74.9	79.6	73.9
0.2	79.3	73.9	80.1	75.6	79.9	74.1
0.3	79.8	78.8	80.8	79.6	80.1	79.1
0.4	82.1	79.2	82.5	80.3	81.3	79.8
0.5	82.9	80.3	83.2	81.5	82.5	81.3
0.6	83.6	81.5	84.2	83.1	82.6	82.5
0.7	83.8	81.9	84.9	82.10	83.6	82.0
0.8	82.1	82.1	84.8	82.4	83.9	81.9
0.9	82.0	81.3	83.6	81.5	82.8	81.0

The experimental results show that  $K_{scr}$  has a better classification performance when the number of training samples is 200, while there is comparable performance when the number of samples is 400. Next we compared the classification accuracy of the graph kernels and RBF corresponding to the 1st- and 2nd datasets for different values of  $\gamma$  used in kernel functions.

**Table 2.** Recognition performance comparison of  $K_{scr}$  with  $K_{OA}$  kernel and RBF for different values of  $\gamma$  used in kernel function. Numbers represent correct classification rate [%].

$\gamma$	Kscr			OA Kernel			RBF		
	DTP	Toxic	Diff.	DTP	Toxic	Diff.	DTP	Toxic	Diff.
0.1	79.	74.9	4.9	79.6	73.9	5.5	78.8	73.8	5.0
0.2	80.1	75.6	4.5	79.9	74.1	5.8	79.3	73.9	5.2
0.3	80.8	79.6	1.2	80.1	79.1	1.0	79.8	78.8	1.0
0.4	82.5	80.3	2.2	81.3	79.8	1.5	82.1	79.2	2.9
0.5	83.2	81.5	1.8	82.5	81.3	1.2	82.9	80.3	2.6

The  $K_{scr}$  shows performance improvements over the RBF and  $K_{OA}$  kernel, for both of the (noisy) real data sets. Moreover it is worth mentioning that  $K_{scr}$  does slightly better than  $K_{OA}$  kernel in general. Finally,  $K_{scr}$  is significantly better than RBF and  $\gamma = 0.7, 0.8$  at a classification rate of 84%. This is a very satisfying result as the definition of activity plays a very important role in modern biostatistics.

We would like now to determine if a new  $K_{scr}$  kernel has an effect in SVM classification in comparison to  $K_{OA}$  and RBF. A permutation test was selected as an alternative way to test for differences in our kernels in a nonparametric fashion (so we do not assume that the population has a normal distribution, or any particular distribution and, therefore, do not make distributional assumptions about the sampling distribution of the test statistics). The R package “exactRankTests” [29] was used for permutation test calculation. Table 3 lists the 9 calculation of SVM accuracy with different  $\gamma$  and results from the test. This Table shows four columns for each pair of compared different kernel methods (both data sets), the first and second giving the classification accuracy, while the last two columns have the raw (i.e., unadjusted) t-statistic result and p-values computed by the resampling algorithm described in [30]. The permutation test based on 2000 sample replacements estimated a p-value to decide whether or not to reject the null hypothesis. The null hypotheses for this test were  $H_{01}: K_{scr} = K_{OA}$ ,  $H_{02}: K_{scr} = RBF$  and alternative hypothesis  $HA_1: K_{scr} > K_{OA}$ ,  $HA_2: K_{scr} > RBF$ , additionally let's assume at a significance level  $\alpha = 0.05$ . The permutation test will reject the null hypothesis if the estimated  $P$ -value is less than  $\alpha$ . More specifically, for any value of  $\alpha < p$ -value, fail to reject  $H_0$ , and for any value of  $\alpha \geq P$ -value, reject  $H_0$ . The  $P$ -values on average for DTS AIDS and toxic data sets of 0.0527, 0.0025, 0.0426, 0.0020 indicates that the SVM with  $K_{scr}$  kernel is probably not equal to  $K_{OA}$  and RBF kernel. The  $P$ -value 0.0527 between  $K_{scr}$  and  $K_{OA}$  kernel for DTS AIDS data sets, indicates weak evidence against the null hypothesis. There is strong evidence that all other tests null hypothesis can be rejected. Permutation tests suggest, on average, that  $K_{scr}$  kernel for screening data is statistically significantly larger than  $K_{OA}$  and RBF.

**Table 3.** Statistical test between  $K_{scr}$ ,  $K_{OA}$  kernel and RBF for different values of  $\gamma$  used in kernel function. Numbers represent correct classification rate [%], t-statistic (without permutation) and calculated p-value from permutation test. Calculated t\*-statistic to the new data set with replacement 2000 times gave result in average  $t_{Min}^* = 0.852$ ,  $t_{Max}^* = 1.254$ .

$\gamma$	DTP AIDS			
	Kscr	OA	t-stat	p-value
0.1	79.8	79.6	3.78	0.0231
0.2	80.1	79.9	4.12	0.0633
0.3	80.8	80.1	2.59	0.0712
0.4	82.5	81.3	3.22	0.0541
0.5	83.2	82.5	2.99	0.0634
0.6	84.2	82.6	-3.12	0.0773
0.7	84.9	83.6	3.02	0.0424
0.8	84.8	83.9	-2.89	0.0561
0.9	83.6	82.8	4.55	0.0233
			Average:	0.0527



## 6 Conclusion

We have developed a new fast kernel function that is suitable for discriminative classification with unordered sets of local features. Our graph burden kernel approximates the optimal partial matching of molecules by computing vertex labels, edge labels and burden values and requires time. The kernel is robust since it does not penalize the presence of extra features, respects the co-occurrence statistics inherent in the input sets, and is provably positive-definite. We have applied our kernel to SVM-based object recognition tasks, and demonstrated recognition performance with accuracy comparable to current methods on screening data. Our experimental evaluation showed that our algorithm leads to substantially better results than those obtained by existing QSAR- and sub-structure-based methods.

## References

1. A. L. Blum, P. Langley, Selection of Relevant Features and Examples in Machine Learning, *Artificial Intelligence*, 1997, 97(12), 245 ± 271.
2. A. Srinivasan, R. D. King, S. H. Muggleton, and M. Sternberg. The predictive toxicology evaluation challenge. In 15th IJCAI, 1997.
3. Burden, F.R. 1989. "Molecular Identification Number For Substructure Searches", *Journal of Chemical Information and Computer Sciences*, 29, 225-227.
4. dtp.nci.nih.gov. Dtp aids antiviral screen dataset.
5. D.P. Mahe, N. Ueda, T. Akutsu, J.-L. Perret and J.-P. Vert, "Extensions of Marginalized Graph Kernels," *Proc. 21st Int'l Conf. Machine Learning*, 2004
6. G. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: *Machine Learning: Proc. of the 11th Intern. Conf.*, 1994, pp. 121 ± 129.
7. Graham W. Richards. Virtual screening using grid computing: the screensaver project. *Nature Reviews: Drug Discovery*, 1:551–554, July 2002.
8. H. Froehlich, J. K. Wegner, A. Zell, *QSAR Comb. Sci.* 2004, 23, 311 – 318.
9. H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, C. Watkins: "Text Classification using String Kernels" (2002) *Journal of Machine Learning Research*, 2.
10. Hawkins, D.M., Young, S.S., and Rusinko, A. 1997. "Analysis of a Large Structure-Activity Data Set Using Recursive Partitioning", *Quantitative Structure Activity Relationships* 16, 296-302.
11. I. Guyon, A. Elisseeff, An Introduction into Variable and Feature Selection, *Journal of Machine Learning Research Special Issue on Variable and Feature Selection*, 2003, 3, 1157 ± 1182.
12. J. Kandola, J. Shawe-Taylor, and N. Cristianini. On the application of diffusion kernel to text data. Technical report, Neurocolt, 2002. NeuroCOLT Technical Report NC-TR-02- 122.
13. K. Palm, P. Stenborg, K. Luthman, P. Artursson, *Pharm. Res.* 1997, 14, 586 – 571.
14. Kashima, H., Tsuda, K., & Inokuchi, A. (2003). Marginalized kernels between labeled graphs. *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 321-328). AAAI Press.
15. Lam, R. 2001. "Design and Analysis of Large chemical Databases for Drug Discovery", Ph.D thesis presented to Department of Statistics and Actuarial Science, University of Waterloo, Canada.
16. N. Nikolova, J. Jaworska, Approaches to Measure Chemical Similarity. *Review, QSAR & Combinatorial Science*, 2003, 22, 9 ± 10.

17. M. D. Wessel, P. C. Jurs, J.W. Tolan, S. M. J. Muskal, *Chem. Inf. Comput. Sci.* 1998, 38, 726–735.
18. Pearlman, R. S. and Smith, K. M. 1998. "Novel software tools for chemical diversity", *Perspectives in Drug Discovery and Design*, 9/10/11, 339-353.
19. S. Kramer, L. De Raedt, and C. Helma. Molecular feature mining in hiv data. In 7th International Conference on Knowledge Discovery and Data Mining, 2001.
20. S. Vishwanathan, A. Smola, Fast Kernels for String and Tree Matching, in: B. Schoelkopf, K. Tsuda, J.-P.
21. Schoelkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA, MIT Press.
22. Shawe-Taylor, J.; Cristianini, N, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.
23. T. Gaertner. A survey of kernels for structured data. *SIGKDD Explorations*, 5(1), 2003. 270-275, 1999.
24. T. Joachims, *Learning to Classify Text using Support Vector Machines: Machines, Theory and Algorithms* (Kluwer Academic Publishers, Boston, 2002).
25. Todeschini, R.; Consonni, V. (Eds.), *Handbook of Molecular Descriptors*, Wiley-VCH, Weinheim, 2000.
26. Tsuda, K., Kin, T., & Asai, K. (2002). Marginalized kernels for biological sequences. *Bioinformatics*, 18, S268–S275.
27. Washio, T., & Motoda, H. (2003). State of the art of graph-based data mining. *SIGKDD Explorations Special Issue on Multi-Relational Data Mining*, 5.
28. [www-ra.informatik.uni-tuebingen.de/software/joelib/](http://www-ra.informatik.uni-tuebingen.de/software/joelib/). Cheminformatics library.
29. <http://cran.r-project.org/src/contrib/Descriptions/exactRankTests.html>. "exactRankTests": Exact Distributions for Rank and Permutation Tests

# A Hybrid Automated Detection System Based on Least Square Support Vector Machine Classifier and $k$ -NN Based Weighted Pre-processing for Diagnosing of Macular Disease

Kemal Polat<sup>1</sup>, Sadık Kara<sup>2</sup>, Ayşegül Güven<sup>3</sup>, and Salih Güneş<sup>1</sup>

<sup>1</sup> Selcuk University, Dept. of Electrical & Electronics Engineering,  
42075, Konya, Turkey

{kpolat, sgunes}@selcuk.edu.tr

<sup>2</sup> Erciyes University, Dept. of Electronics Eng., 38039,  
Kayseri, Turkey

kara@erciyes.edu.tr

<sup>3</sup> Erciyes University, Civil Aviation College,  
Department of Electronics, 38039 Kayseri, Turkey  
sguven@erciyes.edu.tr

**Abstract.** In this paper, we proposed a hybrid automated detection system based least square support vector machine (LSSVM) and  $k$ -NN based weighted pre-processing for diagnosing of macular disease from the pattern electroretinography (PERG) signals.  $k$ -NN based weighted pre-processing is pre-processing method, which is firstly proposed by us. The proposed system consists of two parts:  $k$ -NN based weighted pre-processing used to weight the PERG signals and LSSVM classifier used to distinguish between healthy eye and diseased eye (macula diseases). The performance and efficiency of proposed system was conducted using classification accuracy and 10-fold cross validation. The results confirmed that a hybrid automated detection system based on the LSSVM and  $k$ -NN based weighted pre-processing has potential in detecting macular disease. The stated results show that proposed method could point out the ability of design of a new intelligent assistance diagnosis system.

## 1 Introduction

The pattern electroretinogram (PERG) provides an objective measure of central retinal function, and its importance is increasing daily in clinical visual electrophysiological practice. Simultaneous recording of PERG has been utilized in the last few years to study disease of the macula and optic nerve. Riggs and colleagues first reported the existence of the PERG in 1964 [1], [2]. PERG is the subject of intense laboratory investigation. The demonstration in humans and animals raised clinical interest because of the potential usefulness of noninvasive measurement of ganglion cell activity [3]. More detailed reports of PERGs in optic nerve or macular dysfunction have subsequently appeared [4], [5], [6], [7], [8].

We have used the  $k$ -NN based weighted pre-processing method to weight the real valued PERG signals. Then, we have used the LSSVM classifier, which one of best classification systems in machine learning. For 10, 15, and 20 values of  $k$  to  $k$ -NN based

weighted pre-processing, we classified the macular disease using LSSVM classifier. The obtained classification accuracies are 100% for above each  $k$  value using 10-fold cross validation. While only the LSSVM classifier obtained 90.91% classification accuracy, proposed system obtained 100% classification accuracy for 10, 15, and 20 values of  $k$ .

The remaining of the paper is organized as follows. We present the used PERG signals in the next section. In Sect. 3, we give the used procedure. We present the experimental data and obtained results to show the effectiveness of our method in Sect. 4. Finally, we conclude this paper in Sect. 5 with future directions.

## 2 Acquiring of PERG Signals and Macular Disease

Fifty-six subjects suffering macular disease and 50 healthy volunteer participated in the study. The mean age of the entire group of patients was 44.6 years (23 females and 33 males). And 50 healthy subjects were examined as a control group; were consist of 38 females and 12 males with a mean age of 41 years (SD 5.0).

Healthy volunteers and patients with macula disease gave their written informed consent for participation in the study, which was performed with the approval of the ethics committee of the Erciyes University Hospital (Kayseri, Turkey) and carried out in accordance with the Declaration of Helsinki (1989) of the world medical association.

Electrophysiological test devices were used during examinations and signals were taken into consideration. Recordings of the PERG signals were made using Tomey Primus 2.5 electrophysiology unit (TOMEY GmbH Am Weichselgarten 19a 91058 Erlangen Germany) in the ophthalmology department of Erciyes University Hospital.

The bioelectrical signal was recorded by a small Ag/AgCl skin electrode. The electrode embedded in a contact lens, which is placed on the subject's left cornea. A full field (Ganzfeld) light stimulus is used. The recording electrodes should consist of a corneal contact lens electrode, which supports the eyelids, and reference electrodes placed centrally on the forehead or near each orbital rim. The ground electrode was located on the ear. The stimulus consists of a checkerboard. The computer records the PERG response samples over a period of 204 ms. The time progression of these frames forms a contour, which is examined by a medical expert to determine the presence of eye diseases [1], [7].

Contrast between the alternating checkerboards was set to 80%. Frequency of stimulation was 1.02 /Hz, stimulations per minute was 71.2, acquisition time was 204 ms, number of stimuli was set at 150. Pupils were not dilated. The subjects wore their full spectacle correction without interfering with the electrodes. The high and low band pass input filters were set at 1 and 30 Hz respectively, and the signal was averaged using a computer.

## 3 The Proposed Method

### 3.1 Overview

Figure 1 shows the procedure used in the proposed system. It consists of four parts: (a) Loading of macular disease that has 63 features, (b) Weighted Pre-processing-  $k$ -NN Based Weighted Pre-processing (LSSVM inputs were selected), (c) Classification system (LSSVM classifier) (d) Classification results (macular disease and healthy).

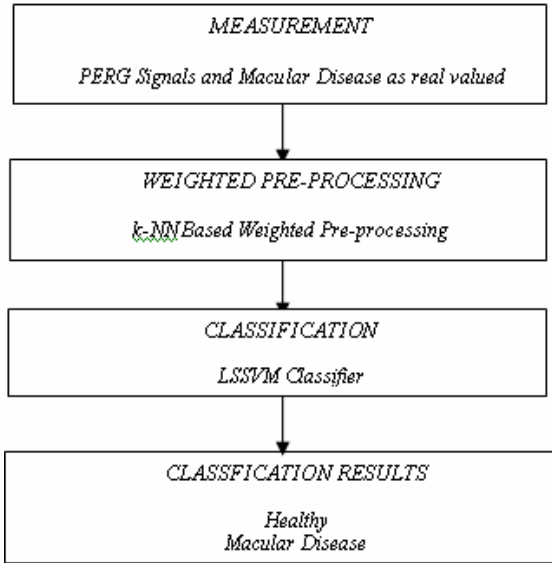


Fig. 1. The used procedure

### 3.2 k-NN Based Weighting Pre-processing

We know from the fuzzy systems that sometimes attributing new values, which are membership values in the fuzzy-logic case, to the data samples can give better results with respect to the classification accuracy. This process is done for each attribute of each data sample through giving a new value according to the membership functions. In this study however, we did this re-determination of attribute values of each data sample by using *k*-NN algorithm in the following manner (Fig. 2).

The weighting process is conducted for each attribute separately. For example, all of the values of the first attribute among the whole data is reconsidered and changed according to the *k*-NN scheme and then the same process is conducted for the second attribute values of the data samples and son on. The process that is done in determining one attribute value of one data sample can be explained simply like this: Let *i* be the data sample label and *j* be the attribute index. That is we are searching a new value for the *j<sup>th</sup>* value of the *i<sup>th</sup>* data sample. As can be seen from the Fig.2, the first thing we must do is to calculate the distances of all other attribute values of the same data sample, *i<sup>th</sup>* data sample in this case, to this attribute value. Here, a simple absolute difference is utilized as a distance measure (1):

$$d(x_i(j), x_i(m)) = |x_i(j) - x_i(m)| \tag{1}$$

where  $x_i(j)$  is the *j<sup>th</sup>* attribute of *i<sup>th</sup>* data sample, while  $x_i(k)$  is the *m<sup>th</sup>* attribute of *i<sup>th</sup>* data sample. After the calculation of the distances, the nearest *k* attribute values are taken and the mean value of these values is calculated (2):

$$mean\_value(j) = \frac{1}{k} \sum_{attr_n \in attr_k, n=1}^k attr_n, \tag{2}$$

where  $attr_n$  represents the value of the  $n^{th}$  attribute in the  $k$ -nearest attribute values and  $k$  is the number of nearest points to the related attribute value  $j$ . This calculated mean value is taken as the new value of the related attribute value for which these calculations are done. That is,  $j^{th}$  value of the  $i^{th}$  data sample is changed with this mean value. The process is conducted for each attribute value of the same data sample in the same manner [13].

```

For each attribute j
  For each data sample i
    *calculate the distances of all other attribute
      values to the attrj,i:
      For n=1:N-1
        d(n)=distn≠j(attrj,i, attrj,n)
      End
    *Find the nearest k attribute values:
      KN1×k←nearest k attribute values
    *Find the mean value of nearest points:
      meanj,i=mean(KN) ;
    *Change the value of attrj,i:
      attrj,i=meanj,i
  next i
next j
    
```

**Fig. 2.**  $k$ -NN based weighting process, where  $N$  is the number of data samples

### 3.3 The Least Square Support Vector Machine Classifier

In this section we firstly mention about SVM classifier after that LSSVM related to SVM.

#### 3.3.1 Support Vector Machines (SVMs)

SVM is a reliable classification technique, which is based on the statistical learning theory. This technique was firstly proposed for classification and regression tasks by [9].

A linear SVM was developed to classify the data set which contains two separable classes such as  $\{+1, -1\}$ . Let the training data consist of  $n$  datum  $(x_1, y_1), \dots, (x_n, y_n)$ ,  $x \in R^n$  and  $y \in \{+1, -1\}$ . To separate these classes, SVMs have to find the optimal (with maximum margin) separating hyperplane so that SVM has good generalization ability. All of the separating hyperplanes are formed with,

$$D(x) = (w * x) + w_0 \tag{3}$$

and provide following inequality for both  $y = +1$  and  $y = -1$

$$y_i [(w * x_i) + w_0] \geq 1, \quad i = 1, \dots, n. \tag{4}$$

The data points which provide above formula in case of equality are called the support vectors. The classification task in SVMs is implemented by using of these support vectors.

Margins of hyperplanes obey following inequality

$$\frac{yk \times D(xk)}{\|w\|} \geq \Gamma, \quad k=1, \dots, n. \tag{5}$$

To maximize this margin ( $\Gamma$ ), norm of  $w$  is minimized. To reduce the number of solutions for norm of  $w$ , following equation is determined

$$\Gamma \times \|w\| = 1. \tag{6}$$

Then formula (7) is minimized subject to constraint (4)

$$1/2 \|w\|^2. \tag{7}$$

When we study on the non-separable data, slack variables  $\xi_i$ , are added into formula (4) and (7). Instead of formulas (4) and (7), new formulas (8) and (9) are used

$$y_i [(w \cdot x_i) + w_0] \geq 1 - \xi_i. \tag{8}$$

$$C \sum_{i=1}^n \xi_i + 1/2 \|w\|^2. \tag{9}$$

Since originally SVMs classify the data in linear case, in the nonlinear case SVMs do not achieve the classification tasks. To overcome this limitation on SVMs, kernel approaches are developed. Nonlinear input data set is converted into high dimensional linear feature space via kernels. In SVMs, following kernels are most commonly used:

- Dot product kernels :  $K(x, x') = x \cdot x'$  ;
- Polynomial kernels :  $K(x, x') = (x \cdot x' + 1)^d$  ; where  $d$  is the degree of kernel and positive integer number;
- RBF kernels:  $K(x, x') = \exp(-\|x - x'\|^2 / \sigma^2)$  ; where  $\sigma$  is a positive real number.

In our experiments  $\sigma$  and  $C$  are selected as 100 and 0.9, respectively.

### 3.3.2 LSSVM (Least Squares Support Vector Machines)

LSSVMs are proposed by [10]. The most important difference between SVMs and LSSVMs is that LSSVMs use a set of linear equations for training while SVMs use a quadratic optimization problem [4]. While formula (9) is minimized subject to formula (8) in Vapnik’s standard SVMs, in LSSVMs formula (11) is minimized subject to formula (10):

$$y_i [(w \cdot x_i) + w_0] = 1 - \xi_i, \quad i=1, \dots, n \tag{10}$$

$$1/2 \|w\|^2 + \frac{C}{2} \sum_{i=1}^n \xi_i^2. \tag{11}$$

According to these formulas, their dual problems are built as follows:

$$Q(w, b, \alpha, \xi) = \frac{1}{2} \|w\|^2 + \frac{C}{2} \sum_{i=1}^n \xi_i^2 - \sum_{i=1}^n \alpha_i \{y_i[(w \cdot x_i) + w_b] - 1 + \xi_i\}. \tag{12}$$

Another difference between SVMs and LSSVMs is that  $\alpha_i$  (lagrange multipliers) are positive or negative in LSSVMs but they must be positive in SVMs. Information in detailed is found in [10] and [11].

## 4 The Experimental Results

In this section, we present the performance evaluation methods used to evaluate the proposed method. Finally, we give the experimental results and discuss our observations from the obtained results.

### 4.1 Classification Accuracy

In this study, the classification accuracies for the datasets are measured using (13):

$$accuracy(T) = \frac{\sum_{i=1}^T assess(t_i)}{|T|}, t_i \in T \tag{13}$$

$$assess(t) = \begin{cases} 1, & \text{if } classify(t) = t.c \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where  $T$  is the set of data items to be classified (the test set),  $t \in T$ ,  $t.c$  is the class of item  $t$ , and  $classify y(t)$  returns the classification of  $t$  by LSSVM classifier.

### 4.2 K-Fold Cross Validation

K-fold cross validation is one way to improve the holdout method. The data set is divided into  $k$  subsets, and the holdout method is repeated  $k$  times. Each time, one of the  $k$  subsets is used as the test set and the other  $k-1$  subsets are put together to form a training set. Then the average error across all  $k$  trials is computed. Every data point appears in a test set exactly once, and appears in a training set  $k-1$  times. The variance of the resulting estimate is reduced as  $k$  is increased. A variant of this method is to randomly divide the data into a test and training set  $k$  different times [12].

### 4.3 Results and Discussion

To evaluate the effectiveness of our method, we made experiments on the PERG signals database mentioned above. Table 1 gives the classification accuracies of our method and LSSVM classifier. As we can see from these results, our method using 10-fold cross validation obtains the highest classification accuracy, 100%.



**Table 1.** Classification accuracies of LSSVM classifier to Macular disease classification problem using the 10-fold cross validation

<i>k</i> value for <i>k</i> -NN based weighted pre-processing	Classification Accuracy (%)
10	100
15	100
20	100
We do not use <i>k</i> -NN based weighted pre-processing	90.91

The obtained results have been shown that proposed *k*-NN based weighted pre-processing method is available for diagnosing of macular disease. As we can be seen from the above results, we conclude that the hybrid diagnostic system based on the *k*-NN based weighted pre-processing and LSSVM classifier obtains very promising results in classifying the possible macular disease patients.

## 5 Conclusions and Future Work

In this paper, a medical decision making system based on *k*-NN based weighted pre-processing and LSSVM classifier was applied on the task of diagnosing macular disease and the most accurate learning methods was evaluated. The results strongly suggest that the hybrid automated detection system based on the *k*-NN based weighted pre-processing and LSSVM classifier can aid in the diagnosis of macular disease. It is hoped that more interesting results will follow on further exploration of data.

**Acknowledgments.** This study has been supported by Scientific Research Project of Selcuk University (Project No: 05401069).

## References

1. Quigley HA, Addicks EM, Green WR.: Optic nerve damage in human glaucoma. III. Quantitative correlation of nerve fiber loss and visual defect in glaucoma, ischemic neuropathy, papilledema and toxic neuropathy. *Arch Ophthalmol* (1982) 100:135–46
2. Quigley HA, Dunkelberger GR, Green WR.: Chronic human glaucoma causing selectively greater loss of large optic nerve fibers. *Ophthalmology* (1988) 95:357–63
3. Bobak P, Bodis-Wollner I, Harnois C, et al.: Pattern electroretinograms and visual-evoked potentials in glaucoma and multiple sclerosis. *Am J Ophthalmol* (1983) 96:72–83
4. Falsini B, Colotto A, Porciatti V, et al.: Macular flicker- and pattern-ERGs are differently affected in ocular hypertension and glaucoma. *Clin Vis Sci* (1991) 6:423–9
5. Graham SL, Wong VAT, Drance SM, Mikelberg FS.: Pattern electroretinograms from hemifields in normal subjects and patients with glaucoma. *Invest Ophthalmol Vis Sci* (1994) 35: 3347–56
6. O'Donoghue E, Arden GB, O'Sullivan F, et al.: The pattern electroretinogram in glaucoma and ocular hypertension. *Br J Ophthalmol* (1992) 76:387–94
7. Pfeiffer N, Tillmon B, Bach M.: Predictive value of the pattern electroretinogram in high-risk ocular hypertension. *Invest Ophthalmol Vis Sci* (1993) 34:1710–5

8. Porciatti V, Falsini B, Brunori S, et al.: Pattern electroretinogram as a function of spatial frequency in ocular hypertension and early glaucoma. *Doc Ophthalmol* (1987) 65:349–55
9. V. Vapnik: *The Nature of Statistical Learning Theory*, Springer, New York (1995)
10. Suykens, J. A. K., & Vandewalle, J., 1999. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3), 293-300
11. Daisuke Tsujinishi, Shigeo Abe: Fuzzy least squares support vector machines for multi-class problems, *Neural networks field*, Elsevier, 16 (2003) 785-792
12. Kohavi, R, and Provost, F., (1998). Glossary of Terms. Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process, 30, Number 2/3
13. Polat, K, and Güneş, S.: A hybrid medical decision making system based on principles component analysis, k-NN based weighted pre-processing and adaptive neuro-fuzzy inference system, *Digital Signal Processing*, 16:6 (2006) 913-921

# Analysis of Microscopic Mast Cell Images Based on Network of Synchronised Oscillators

Michal Strzelecki<sup>1,2</sup>, Hyongsuk Kim<sup>2</sup>, Pawel Liberski<sup>3</sup>, and Anna Zalewska<sup>4</sup>

<sup>1</sup> Institute of Electronics, Technical University of Lodz,  
Wolczanska 211/215, 90-924 Lodz, Poland  
michal.strzelecki@p.lodz.pl  
<http://www.elel.p.lodz.pl>

<sup>2</sup> Division of Electronics and Information Engineering,  
Chonbuk National University, 561-756 Jeonju, Korea  
hskim@chonbuk.ac.kr

<sup>3</sup> Molecular Biology Department, Medical University of Lodz,  
Czechoslowacka 8/10, 92-216 Lodz, Poland  
ppliber@csk.am.lodz.pl

<sup>4</sup> Department of Dermatology, Medical University of Lodz,  
Krzemieńska 5, 94-014 Lodz, Poland  
dermatol@csk.umed.lodz.pl

**Abstract.** This paper describes automatic analysis of microscopic mast cell images using network of synchronized oscillators. This network allows for detection of image objects and their boundaries along with evaluation of some geometrical object parameters, like their area and perimeter. Estimation of such mast cells' parameters is very important in description of both physiological and pathological processes in the human organism. It was also demonstrated, that oscillator networks is able to perform basic morphological operations along with binary image segmentation. Analysis of sample mast cell image was presented and discussed.

## 1 Introduction

Mast cells are key important cellular elements of the skin and internal organs. It is well established that they take an active part in many immunological processes. Secretory granules of mast cells are the most characteristic morphological structure of those cells [4]. They contain many vital substances including histamine, proteoglycans, prostaglandines and numerous cytokines, which take a crucial part in any physiological and pathological processes in the organism. Mast cells are employed in the pathogenesis of both allergic diseases and non-allergic ones including mastocytosis, in which they are key elements responsible for this disease development, scleroderma, psoriasis, lichen planus, porphyria cutanea tarda, neurofibromatosis, and many others. Detailed morphological analysis of mast cell granules seems to be very important in estimation of mast cell activation [12].

The aim of this study was to perform the image analysis of mast cell granules visualised by electron microscopy. They were obtained from the skin biopsies of healthy volunteers. Calculated cell parameters, like their number, mean perimeter and area in comparison to the whole cell area suggest role importance of those structures in

activation of mast cells and points out at the role of mast cell themselves in physiology of the human organism. To calculate mast cells' geometrical parameters, they were segmented from the image background. Image segmentation into disjoint homogeneous regions is as a very important stage of image processing; it is usually considered that segmentation results strongly influence further image analysis.

Segmentation method presented in this paper implements network of synchronized oscillators [11]. This recently developed tool is based on "temporary correlation" theory [10], which attempts to explain scene recognition, as it would be performed by a human brain. This theory assumes that different groups of neural cells encode different properties of homogeneous image regions (e.g. shape, colour, texture). Monitoring of temporal activity of cell groups allows detection of such image regions and consequently, leads to scene segmentation. Oscillator network was successfully used for segmentation of many biomedical images, e.g. MR brain images [5], MR foot cross-sections textures [6], ultrasound images of intracardiac masses [8]. The advantage of this network is its adaptation to local image changes (related both to the image intensity and texture), which in turn ensures correct segmentation of noisy and blurred image fragments. Another advantage is that synchronized oscillators do not require any training process, unlike the artificial neural networks. Oscillator network is also able to detect object and texture boundaries [7]. In this paper it was also demonstrated, that oscillator network can perform binary morphological operations (along with the segmentation), which are very useful in image preprocessing. Finally, such a network can be manufactured as a CMOS VLSI chip, for very fast image segmentation [2].

## 2 Network of Synchronized Oscillators

To implement the image segmentation technique based on temporary correlation theory, an oscillator network is used. Each oscillator in the network is defined by two differential equations [1], [11]:

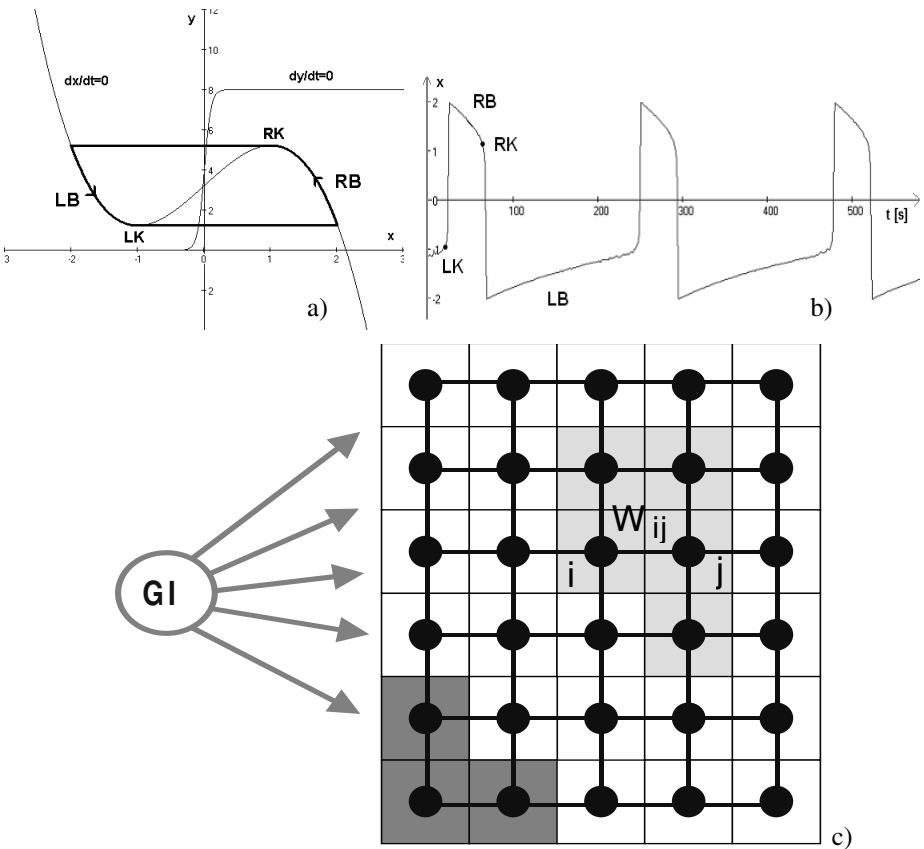
$$\frac{dx}{dt} = 3x - x^3 + 2 - y + I_T, \quad \frac{dy}{dt} = \varepsilon \left[ \gamma \left( 1 + \tanh \left( \frac{x}{\beta} \right) \right) - y \right], \quad (1)$$

where  $x$  is referred to as an excitatory variable while  $y$  is an inhibitory variable.  $I_T$  is a total stimulation of an oscillator and  $\varepsilon, \gamma, \beta$  are parameters. The  $x$ -nullcline is cubic curve while the  $y$ -nullcline is a sigmoid function as shown in Fig. 1a. If  $I_T > 0$ , then equation (1) possesses periodic solution, represented by bold black line shown in Fig. 1a. The operating point moves along this line, from left branch (LB - it represents so-called silent phase), then jumps from left knee (LK) to right branch (RB - it represents so-called active phase), next reaches right knee (RK) and jumps again to left branch. Waveform of an active oscillator is presented in Fig. 1b. If  $I_T \leq 0$ , the oscillator is inactive (produces no oscillation). Oscillators defined by (1) are connected to form a two-dimensional network. In the simplest case, each oscillator is connected only to its four nearest neighbours (larger neighbourhood sizes are also possible). Sample network is shown in Fig. 1c. Network dimensions are equal to dimensions of the analyzed image and each oscillator represents a single image pixel. Each oscillator in the network is connected with so-called global inhibitor (GI in Fig. 1c), which receives

information from oscillators and in turn eventually can inhibit the whole network. Generally, the total oscillator stimulation  $I_T$  is given by the equation:

$$I_T = I_{in} + \sum_{j \in N(i)} W_{ij} H(x_j - \theta_x) - W_z z, \tag{2}$$

where  $I_{in}$  represents external stimulation to the oscillator (image pixel value).  $W_{ij}$  are synaptic weights, which connect oscillator  $i$  and  $j$ . The number of these weights neighbourhood size  $N(i)$ . In the case of the network in Fig. 1b,  $N(i)$  contains four nearest neighbours of the  $i$ -th oscillator (except for these located on network boundaries). Due to these local excitatory connections, an active oscillator spreads its activity over the whole group of oscillators that represents an image object. It provides a synchronization of the whole group.  $\theta_x$  is a threshold, above which oscillator  $k$  becomes



**Fig. 1.** Nullclines and trajectory of eq. (1) (a) and oscillator waveform  $x(t)$  (b). A fragment of oscillator network (c). Each oscillator is connected with its neighbors using positive weights  $W_{ij}$ . Global inhibitor (GI) is connect to each oscillator.

active.  $H$  is a Heaviside function, it is equal to one if its argument is higher than zero and zero otherwise.  $W_z$  is a weight (with negative value) of inhibitor  $z$ , which is equal to one if at least one network oscillator is in active phase ( $x > 0$ ) and it is equal to zero otherwise. The role of global inhibitor is to provide desynchronisation of oscillator groups representing different objects from the one, which is synchronized at the moment. Global inhibitor will not affect any synchronized oscillator group because the sum in (2) has a greater value than  $W_z$ .

In case of segmentation of binary images, network weights are set according to equation (3):

$$W_{ij} = \begin{cases} W_T & \text{if } I_i = I_j \\ 0 & \text{otherwise} \end{cases}, \quad (3)$$

where  $I_i, I_j$  correspond to grey levels of image points connected to oscillators  $i$  and  $j$  respectively,  $W_T$  is a constant. Hence, weights are high for image objects (and background) and low for object boundaries. Because excitation of any oscillator depends on the sum of weights of its neighbours, all oscillators in the image object oscillate in synchrony. A different oscillator group represents each such an object. Oscillator's activation is switched sequentially between groups in such a way that at a given time only one group (representing given object) is synchronously oscillating. Detection of image objects is performed by analysis of oscillators' outputs [1], [9].

It is also possible to detect image object boundaries by means of the oscillator network. For this purpose, only oscillators related to object border pixels should be activated. Therefore weight connecting oscillators  $i$  and  $j$  should be set as follows:

$$W_{ij} = \begin{cases} W_T & \text{if } I_i \neq I_j \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

In this case weights will be large on object boundaries (large difference between grey level  $I_i$  and  $I_j$ ) and only oscillators located there will be activated. Active oscillators delineate an edge of a given image object and analysis of their outputs allows region boundary detection.

A segmentation algorithm using oscillator network was presented in [3]. It is based on simplified oscillator model and does not require solution of (1) for each oscillator. This algorithm was used to perform segmentation of several biomedical images, as presented in [5], [6], [7], [8].

### 3 The Image Analysis Algorithm

In order to segment mast cells and calculate their number and selected geometrical parameters the following image analysis was performed:

1. thresholding of original grey level image to obtain binary image,
2. removing of small artefact objects,
3. morphological closing operation to separate cells which could be connected after thresholding,
4. segmentation of mast cells, calculation of their number, perimeter and area.

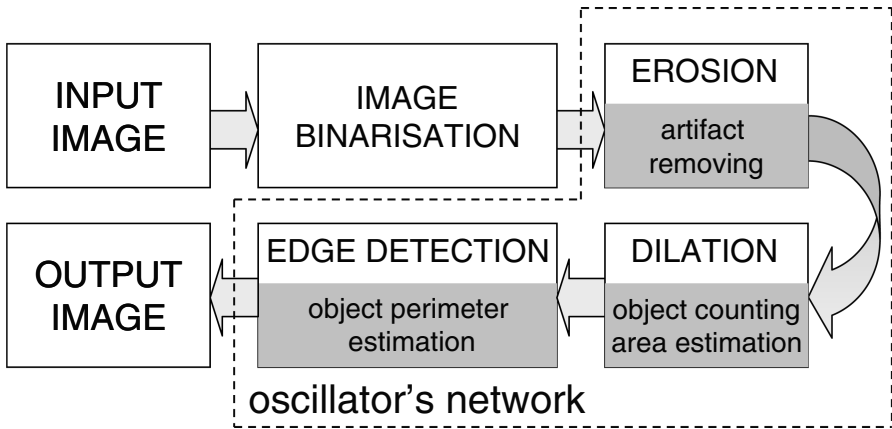


Fig. 2. Image analysis steps

These analysis steps presents block diagram shown in Fig. 2. Stages 2-4 are performed by oscillators’ network. These operations require three passes of the network operation, where oscillator weights are set in different manner. In the first pass, artefacts are removed and image erosion is performed. In the second network pass the dilatation is executed along with mast cell segmentation and calculation of cell number and their area. In the third pass, for the previously obtained image, object boundaries are detected. This allows mast cell perimeter calculation. Cells area and perimeter are evaluated based on number of active oscillators, which belong to detected image objects. Network algorithm is described as follows [9]:

**Step 1**

*Pass 1. Artefact reduction and image erosion*

1.1 Set weights  $W_{ij} = W_T$  for oscillators representing image points  $I_i$  and  $I_j$  when both image points belong to an objects and for each of them the following equation is satisfied:

$$\sum_s I_{i+s} E_s = N, \tag{5}$$

where  $E$  is structuring erosion element,  $N$  is a number of its non-zero elements, and  $W_T$  is a positive constant,  $I$  and  $s$  represent image and element  $E$  sites, respectively. If (5) is not satisfied, weights  $W_{ij}$  are set to zero. This procedure guaranties that oscillation propagation will stop on the oscillators connected with image points, for those the structuring element is contained within the object. This results in object erosion. The structuring element  $E$  used in this study is shown in Fig. 3a.

1.2 Find the leaders. To do this the following equation is considered:

$$\sum_s I_{i+s} M_s = N, \tag{6}$$

where  $M$  is a binary rectangular mask (shown in Fig. 3b), which defines a minimal object to be recognised as a valid mast cell,  $N$  is number of non-zero mask elements,  $i$  and  $s$  represent corresponding image and mask sites, respectively. Central (middle) point of the mask  $M$  is moved over whole image and for each image point  $i$  eq. (6) is evaluated. If it is satisfied, the oscillator connected with point  $i$  is considered as a leader. This means, that objects smaller than defined by  $M$  will not contain leader points, thus they will never oscillate [11]. As a consequence, such an object will be not recognised and segmented.

*Pass 2. Image dilatation and segmentation (calculation of cell number and their area)*

1.1 Set weights  $W_{ij} = W_T$  for oscillators representing image points  $I_i$  and  $I_j$  when for both of them the following equation is satisfied:

$$\sum_s I_{i+s} E_s \neq 0. \quad (7)$$

Otherwise weights  $W_{ij}$  are set to zero. In this case, oscillators will be connected with positive weights if image points they represent and centrally located element  $E$  have at least one common point with the image object. This network operation will result in dilatation for the whole image.

1.2 Find the leaders. In pass 2 each object point is considered as the leader. Set an object and area counters to zero.

*Pass 3. Detection of objects boundaries and perimeter calculation*

1.1 Set weights  $W_{ij}$  of oscillators representing image points  $I_i$  and  $I_j$  according to formula (4).

1.2 Find the leaders. In this pass each object point is considered as the leader. Set an object and perimeter counters to zero.

*Common part for all three network's passes*

1.3 Initialise all oscillators by randomly setting their starting points on Left Branch (LB, see Fig. 1a).

**Step 2.** For each leader calculate the time  $t_i$  needed to reach Left Knee (LK, see Fig. 1a).

**Step 3.** Find the oscillator  $m$  with the shortest time  $t_m$  to LK. Move current position of each oscillator toward LK by the time interval  $t_m$ . Jump oscillator  $m$  to Right Branch. In pass 2, increase the object and area/perimeter counters.

**Step 4. For each oscillator  $i$  (except  $m$ ) do the following:**

4.1 Update its nullcline by calculation of its  $I_T$  (using equation (2))

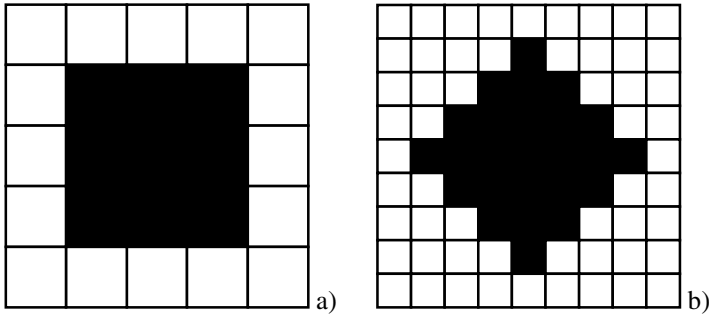
4.2 If oscillator  $i$  is at or beyond its updated knee, then  $i$  jumps to RB

Repeat Step 4 until no further jumping occurs. In pass 2, if jump occurred, increase the object area counter. In pass 3, if jump occurred, increase the object perimeter counter.



**Step 5.** Mark all leaders, which jumped to RB. Force all oscillators on RB back to LB.

**Step 6.** Go to Step 2, considering only these leaders, which are not marked.



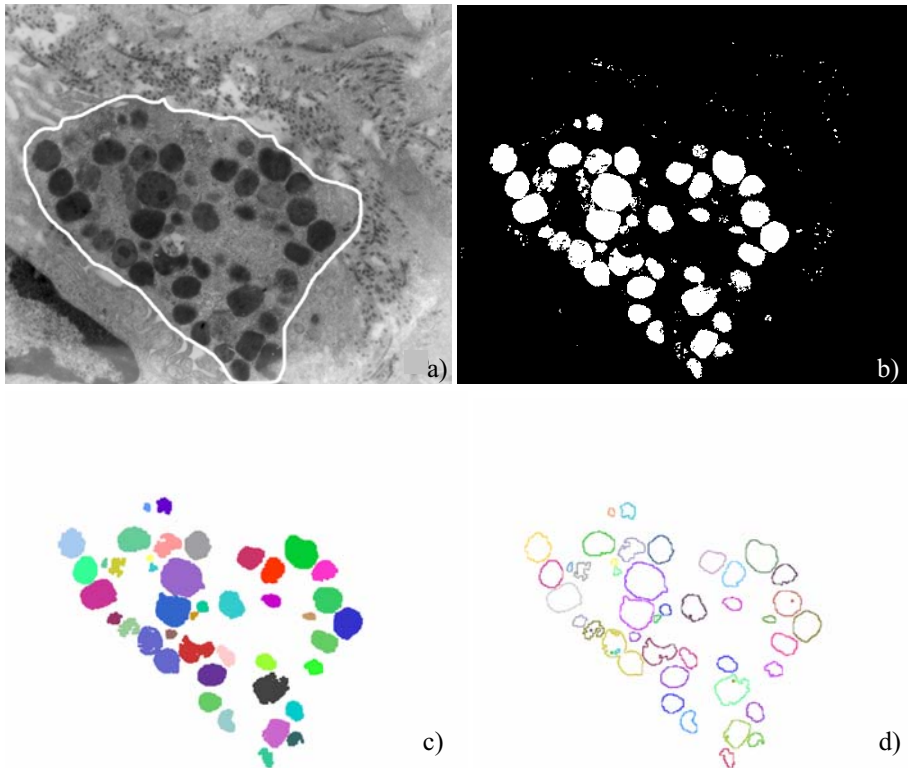
**Fig. 3.** Structuring element  $E$ :  $N=9$  (a), mask  $M$  ( $N=25$ ) (b)

### 4 Computer Simulations

Sample microscopic image, which contains mast cells, is shown in Fig 4a (the whole cell boundary is marked with white line). Its binarised version after removing side objects is presented in Fig. 4b. The segmentation results are shown in Fig. 4c,d. Majority of objects was detected correctly. Some segmentation errors are related to holes, resulted by thresholding and not removed by morphological closing and still present in a few detected mast cells. Image characteristics like area and perimeter were obtained during segmentation by calculating the number of active network oscillators, thus no further image analysis is needed. All calculated image characteristics, useful for medical diagnosis, are presented in Table 1. The area of whole cell was calculated separately. Calculation of cell area is biased by earlier mentioned holes, which could also influence perimeter estimation. However, for calculation of mean cell perimeter, these artefact objects were not taken into consideration (they were skipped during third pass of the network operation after comparison of their perimeter to an arbitrarily assumed minimal perimeter of a valid object).

**Table 1.** Calculated image characteristics

Image characteristics	Fig 4a
Area of the whole cell [pixels]	70084
Number of mast cells	43
Total area of mast cells [pixels]	21001
Mean cell area [pixels]	488.4
Mean cell perimeter [pixels]	97.9
Mean cell shape coefficient ( $\text{perimeter}^2/4\pi \text{ area}$ )	1.56



**Fig. 4.** Sample mast cell microscopi image (a), the same image after thresholding (b), image after analysis: detection of whole cells for number and area calculation (c), detection of cell boundaries for perimeter estimation (d)

## 5 Conclusion

It was demonstrated, that presented segmentation method could be used for analysis of mast cell images. Oscillator network can provide image preprocessing, which contains artefact removing and morphological operations, followed by image segmentation. This segmentation, depending of network weights settings can lead to detection of whole objects or objects boundaries. Calculation of number of active network oscillators ensures evaluation of some objects geometrical characteristics like their number, area and perimeter. This is done parallel to image segmentation without any additional computational effort. The analysis time for the image from Fig. 4a (492×445 pixels) was about 12s using 1.6 GHz P4-based PC. Application of separate algorithms for morphological opening and geometrical parameters estimation took about 15s using the same microcomputer.

Discussed oscillator network was also realized as an ASIC VLSI chip [2]. Currently, network chip is under functional tests. It is expected, that this hardware

realization will significantly reduce the time of network processing due to parallel oscillators operation for each image point, if compared to computer simulations.

**Acknowledgements.** This work was supported by the MIC (Ministry of Information and Communication), Republic of Korea, under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Assessment).

## References

1. Çesmeli, E., Wang, D.: Texture Segmentation Using Gaussian-Markov Random Fields and Neural Oscillator Networks. *IEEE Trans. on Neural Networks* 12 (2001) 394-404
2. Kowalski, J., Strzelecki, M.: CMOS VLSI Chip for Segmentation of Binary Images. *Proc. of IEEE Workshop on Signal Processing, Poznan, Poland (2005)* 251-256
3. Linsay, P., Wang, D.: Fast numerical integration of relaxation oscillator networks based on singular limit solutions. *IEEE Trans. on Neural Networks*, 9 (1998) 523-532
4. Madeja, Z., Korohoda, W.: Some applications of computer image analysis in cell biology. *Post. Biol. Kom.* 23 (1996) 457-476
5. Shareef, N., Wang, D., Yagel, R.: Segmentation of Medical Images Using LEGION. *IEEE Trans. on Med. Imag.* 18 (1999) 74-91
6. Strzelecki, M.: Segmentation of MRI trabecular-bone images using network of synchronised oscillators. *Machine Graphics & Vision* 11 (2002) 77-100
7. Strzelecki, M.: Texture boundary detection using network of synchronized oscillators. *El. Letters* 40 (2004) 466-467
8. Strzelecki, M., Materka, A., Drozd, J., Krzeminska-Pakula, M., Kasprzak, J.: Classification and segmentation of intracardiac masses in cardiac tumor echocardiograms. *Comp. Med. Imag. and Graphics* 30 (2006) 95-107
9. Strzelecki, M.: Image texture segmentation using oscillator networks and statistical methods. *Scientific Letters* no 946. Technical University of Lodz (2004)
10. Wang, D., Ternan, D.: Locally excitatory globally inhibitory oscillators network. *IEEE Trans. on Neural Networks* 6 (1995) 283-286
11. Wang, D., Ternan, D.: Image segmentation based on oscillatory correlation. *Neural Computation* 9 (1997) 805-836
12. Zalewska, A., Strzelecki, M., Sygut, J.: Implementation of an image analysis system for morphological description of skin mast cells in urticaria pigmentosa. *Med. Sci. Monit* 3 (1997) 260-265

# Detection of Gene Expressions in Microarrays by Applying Iteratively Elastic Neural Net\*

Máx Chacón<sup>1</sup>, Marcos Lévano<sup>2</sup>, Héctor Allende<sup>3</sup>, and Hans Nowak<sup>4</sup>

<sup>1</sup> Universidad de Santiago de Chile; Depto. de Ingeniería Informática,  
Avda. Ecuador No 3659 - Casilla 10233; Santiago - Chile  
mchacon@diinf.usach.cl

<sup>2</sup> Universidad Católica de Temuco; Escuela de Ingeniería Informática,  
Avda. Manuel Montt No 56 - Casilla 15-D; Temuco - Chile  
mlevano@uct.cl

<sup>3</sup> Universidad Técnica Federico Santa María; Depto. de Informática  
<sup>4</sup> Depto. de Física

Avda. España No 1680 - Casilla 110-V; Valparaíso - Chile  
hallende@inf.utfsm.cl, hans.nowak@experimentos.cl

**Abstract.** DNA analysis by microarrays is a powerful tool that allows replication of the RNA of hundreds of thousands of genes at the same time, generating a large amount of data in multidimensional space that must be analyzed using informatics tools. Various clustering techniques have been applied to analyze the microarrays, but they do not offer a systematic form of analysis. This paper proposes the use of Gorban's Elastic Neural Net in an iterative way to find patterns of expressed genes. The new method proposed (Iterative Elastic Neural Net, IENN) has been evaluated with up-regulated genes of the Escherichia Coli bacterium and is compared with the Self-Organizing Maps (SOM) technique frequently used in this kind of analysis. The results show that the proposed method finds 86.7% of the up-regulated genes, compared to 65.2% of genes found by the SOM. A comparative analysis of Receiver Operating Characteristic (ROC) with SOM shows that the proposed method is 11.5% more effective.

## 1 Introduction

Modern deoxyribonucleic acid (DNA) microarray technologies [1] have revolutionized research in the field of molecular biology by enabling the study of hundreds of thousands of genes simultaneously in different environments [1].

By using image processing methods it is possible to obtain different levels of expression of thousands of genes simultaneously for each experiment. In this way these techniques generate thousands of data represented in multidimensional space. The process is highly contaminated with noise and subject to measurement errors, finally requiring experimental confirmation. To avoid repeating the whole process experimentally gene by gene, pattern recognition techniques are applied that make it possible to select sets of genes that fulfil given behavior patterns at their gene expression levels.

---

\* This work was supported by projects FONDECYT 1050082, FONDECYT 1040354, FONDECYT 1040365 and MILENIO P02-054-F, Chile.

The most widely used method to determine groupings and select patterns in microarrays is the Self-Organizing Maps (SOM) technique [2], [3], [4]. One of the problems of SOM is the need to have an initial knowledge of the size of the net to project the data, and this depends on the problem that is being studied. On the other hand, since SOM is based on local optimization, it presents great deficiencies by restricting data projections only to its nodes.

One of the recent methods, consensus clustering [5], uses new resampling techniques which should give information about the stability of the found clusters and confidence that they represent real structure. This method is not used in this paper, but will be used and analyzed in a future contribution.

The Elastic Neural Net (ENN) [6], [7] method generates a controllable net described by elastic forces that are fitted to the data by minimizing an energy functional, without the need of knowing its size a priori. This generates greater flexibility to adapt the net to the data, and like the SOMs it allows a reduction in dimensionality, that improves the visualization of the data, which is very important for bioinformatics applications.

ENNs have been applied to different problems in genetics, such as analysis of base sequence structures (adenine, cytosine, guanine and thymine), where base triplet groupings are discovered [7]; automatic gene identification in the genomes of the mitochondria of different microorganisms [8]. But as far as we can tell, there is no application for finding patterns in microarrays.

This paper proposes the use of IENN to divide clusters iteratively, together with the k-means method and using indices to measure the quality of the clusters, making it possible to select the number of groups formed in each iteration.

To evaluate the results, data from the most widely studied microorganism, the bacterium *Escherichia Coli* (*E.Coli*), were used. The levels of gene expression of a set of 7,312 genes were analyzed by means of the microarrays technique. In this set there are 345 up-regulated genes that have been tested experimentally [9] and must be detected with the new method. The results are compared with those of the traditional SOM method.

## 2 Method

### 2.1 Theoretical Foundation

Gorban defines the Elastic Neural Net [6] as a net of nodes or neurons connected by elastic forces (springs), where  $Y = \{y^i, i = 1..p\}$  is a collection of nodes,  $E = \{E^{(i)}, i = 1..s\}$  is a collection of edges, and  $R^{(i)} = \{E^{(i)}, E^{(k)}\}$  is the combination of pairs of adjacent edges called ribs denoted by  $R = \{R^{(i)}, i = 1..r\}$ . Each edge  $E^{(i)}$  starts at node  $E^{(i)}(0)$  and ends at node  $E^{(i)}(1)$ . The ribs start at node  $R^{(i)}(1)$  and end at node  $R^{(i)}(2)$ , with a central node  $R^{(i)}(0)$ . The data to be analyzed are  $x^j = [x_1^j, \dots, x_M^j]^T \in R^M$ , where  $M$  is the dimension of the multidimensional space and  $j = 1..N$  is the number of data.

The set of data closest to a node is defined as a taxon,  $K^i = \{x^j : \|x^j - y^i\| \rightarrow \min\}$ . It is clear that there must be as many taxons as nodes. Here  $\|x^j - y^i\|$  is the norm of the vector  $(x^j - y^i)$ , and the Euclidian norm is used. This means that the taxon  $K^i$  contains all the vectors of the  $x^j$  data whose norms with respect to node  $y^i$  are the smallest.

Energy  $U^{(Y)}$  between the data and the nodes is defined by (1),

$$U^{(Y)} = \frac{1}{N} \sum_{i=1}^p \sum_{x^j \in K^i} \|x^j - y^i\|^2, \tag{1}$$

where each node interacts only with the data of its taxon. An elastic energy between the nodes  $U^{(E)}$  is added by (2),

$$U^{(E)} = \sum_{i=1}^s \lambda_i \|E^i(1) - E^i(0)\|^2, \tag{2}$$

where  $\lambda_i$  are the elasticity constants that allow the net’s elasticity to be controlled. Additionally, a deformation energy  $U^{(R)}$  between pairs of adjacent nodes, is also added by (3),

$$U^{(R)} = \sum_{i=1}^R \mu_i \|R^i(1) - 2R^i(0) + R^i(2)\|^2, \tag{3}$$

where  $\mu_i$  are the deformability constants of the net. The same values of  $\lambda$  and  $\mu$  are chosen for all the  $\lambda_i$  and  $\mu_i$ . The total energy is now minimized by (4) with respect to the number and position of the  $y^i$  nodes for different  $\mu$  and  $\lambda$

$$U = U^{(Y)} + U^{(E)} + U^{(R)}. \tag{4}$$

We used the VIDAEXPERT implementation, which can be found in Gorban et al. [6].

In addition to the flexibility offered by the ENNs to fit the net to the data, the projections of the data to the net can be made over the edges and at points within the net’s cells, and not only over the nodes as required by the SOMs. This leads to an approximation that has a better fit with the real distribution of the data in a smaller space. This property is very important for applications in bioinformatics, where the specialist has better feedback from the process. The same could be said for image processing where the ENN seems to describe well active contours [10].

## 2.2 IENN Method

The algorithm used to find groups of genes that have the same behavior patterns consists of four fundamental phases: data preprocessing, ENN application, pattern identification, and finally a stopping criterion and cluster selection based on the level of expression and inspection of the pattern that is being sought.

### Phase 1: Preprocessing

The set of  $N$  data to be analyzed is chosen,  $x^j = [x^j_1, \dots, x^j_M]^T$ ,  $j = 1 \dots N$ , where  $M$  is the dimension of the multidimensional space. For this application,  $N$  corresponds to the 7,312 genes of the E.coli bacterium and  $M$  to the 15 different experiments carried out on the genes, and  $x^j$  is the gene expression level. The data are normalized in the form  $\theta^j = \ln(x^j - \min(x^j) + 1)$  which is used as a standard in bioinformatics [11].

**Phase 2: Elastic Neural Net (ENN)**

The package of Gorban et al. [6], which uses the following procedures, is applied:

- (a) The data to be analyzed are loaded.
- (b) The two-dimensional net is created according to an initial number of nodes and elastic and deformability constants  $\lambda$  and  $\mu$  with values between 2 for rigid grids and 0.01 for soft grids.
- (c) The net is fitted to the data, minimizing the energy  $U$ . For that purpose the initial values of  $\lambda$  and  $\mu$  are reduced three times (four pairs of parameters are required to be entered by the user). The decrease of  $\lambda$  and  $\mu$  results in a net that is increasingly deformable and less rigid, thereby simulating annealing, allowing the final configuration of the ENN to correspond to an overall minimum of  $U$  or a value very close to it [6].
- (d) The data are projected over the net on internal coordinates. In contrast with the SOM, in which piecewise constant projecting of the data is used (i.e., the data are projected on the nearest nodes), in this method piecewise linear projecting is applied, projecting the data on the nearest point of the net [6]. This kind of projection results in a more detailed representation of the data.
- (e) Steps (c) and (d) are repeated for different initial values of the nodes,  $\lambda$  and  $\mu$ , until the best resolution of the patterns found is obtained.

**Phase 3: Pattern identification**

The data are analyzed by projecting them on internal coordinates for the possible formation of clusters or other patterns such as accumulation of clusters in certain regions of the net. As a typical dependence of the data in a cluster on the dimensions of the multidimensional space, the average of the data for each dimension is calculated (cluster's centroid for the dimension).

For the formation of possible clusters the k-means method is used together with the quality index  $I$  [12], which gives information on the best number of clusters. The centroids of each cluster are graphed and analyzed to find possible patterns.

**Phase 4: Cluster analysis**

Once the best number of clusters is obtained, the centroids' curves are used to detect and extract the possible patterns. In general, the centroid curve of a cluster may present the pattern sought, may be a constant, or may not show a definite trend. Also, the values of the curve can be in a range that is outside the interest of possible patterns (low levels of expression). To decide if the clusters found in a first application of the ENN contain clear patterns, the behavior of the centroids' curves are analyzed. If the centroids' levels are outside the range sought, the cluster is discarded; if the patterns sought are detected, the cluster that contains the genes sought will be obtained (in both cases the division process is stopped), otherwise phases 2 and 3 are repeated with each of the *clusters* and the analysis of phase 4 is carried out again, repeating the process.

**2.3 Data Collection**

The data correspond to the levels of gene expression of 7,312 genes obtained by the microarray technique of E.Coli [9]. These data are found in the GEO database (Gene Expression Omnibus) of the National Center for Biotechnology Information<sup>1</sup>. The

<sup>1</sup> <http://www.ncbi.nlm.nih.gov/projects/GEO/goes>

work of Liu et al. [9] provides the 345 up-regulated genes that were tested experimentally. Each gene is described by 15 different experiments (which correspond to the dimensions for the representation of each gene) whose gene expression response is measured [9] on glucose sources. Specifically there are 5 sources of glucose, 2 sources of glycerol, 2 sources of succinate, 2 sources of alanine, 2 sources of acetate, and 2 sources of proline. The definition of up-regulated genes according to [9] is given in relation to their response to the series of sources of glucose considering two factors: that its level of expression is greater than 8.5 on a  $\log_2$  scale, and that its level of expression increases at least 3 times from the first to the last experiment on the same scale. For our evaluation we considered a less restrictive definition that includes the genes that have only an increasing activity of the level of expression with the experiments; since the definition given in [9] for up-regulated genes contains very elaborate biological information for which a precise identification of the kind of gene to be detected is required.

The original data have expression level values between zero and hundreds of thousands. Such an extensive scale does not offer an adequate resolution to compare expression levels; therefore a logarithmic normalization is carried out. In this case we preferred to use the natural logarithm [11] instead of the base 2 logarithm used by Liu, because it is a more standard measure. The limiting value for the expression level was calculated using our own algorithm by determining the threshold as the value that best separates the initial clusters ( $\theta_{\min}$ ). This expression level allows discarding groups of genes that have an average level lower than this value.

### 3 Results

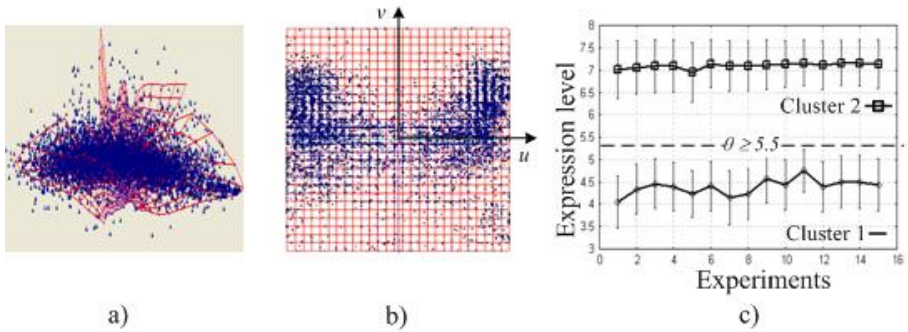
First, the net's parameters were calibrated, i.e. the size of the net was set and the series of pairs of elasticity ( $\lambda$ ) and deformability ( $\mu$ ) parameters were selected. The strategy chosen consisted in evaluating different net sizes and pairs of parameters  $\lambda$  and  $\mu$  for the total data set that would allow minimizing the total energy  $U$ .

The minimum energy was obtained with a mesh of 28x28 nodes that was used throughout the whole process. Implementation of the ENN [6], [7] requires a set of at least four pairs of  $\lambda$  and  $\mu$  parameters to carry out the process, because it adapts the mesh's deformation and elasticity in a process similar to simulated annealing that allows approximation to overall minimums. The set of parameters that achieved the lowest energy values had  $\lambda$  with values of {1.0; 0.1; 0.05; 0.01} and  $\mu$  with values of {2.0; 0.5; 0.1; 0.03}. For the process of minimizing the overall energy  $U$ , 1,000 iterations were used. Then the cluster subdivision iteration process was started.

Figure 1 shows the representation of the first division and the expression levels of the centroids for the two clusters selected by the index  $I$  (for this first iteration). The expression level value equidistant from the two clusters corresponds to  $\theta_{\min}=5.5$ .

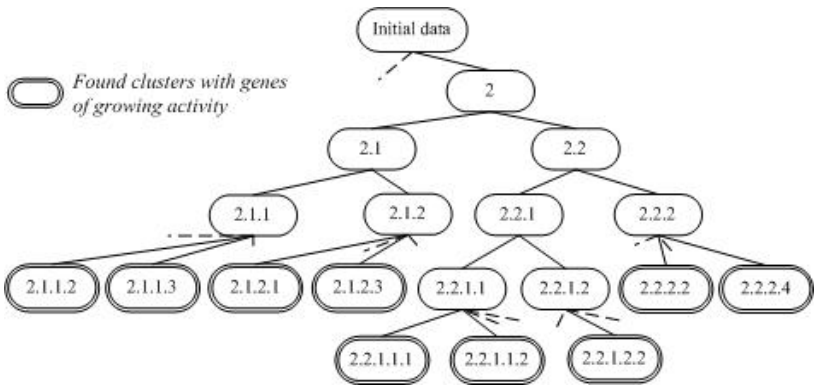
The iteration process generates a tree where each node has branches to a number of subclusters found by the maximum value of the index  $I$ . In the particular case of E.Coli, a tree of depth five is generated. The generation of the tree is made together with a pruning by expression level, i.e., only those clusters that present an expression level greater than  $\theta_{\min} \geq 5.5$  are subdivided.





**Fig. 1.** First iteration of the method. a) Fitted net to original data. b) Projections on internal coordinates. c) Centroids and choice of the minimum expression level.

Finally, to stop the subdivision process of the groups that have an expression level greater than  $\theta_{\min}$ , the behavior of the expression level in the experiments was examined. In this case we only looked for a simple increasing pattern of the expression level in the centroids through the 15 experiments (the strict definition of up-regulated genes given in [9] was not used). Figure 2 shows the tree of subclusters generated by the process applied to the genes of E.Coli.



**Fig. 2.** Prepruned subcluster generation tree. Every leaf shown contains the set of genes with increasing activity where the up-regulated genes to be evaluated are found. The coding of each node shows the sequence of the nodes through which one must go to reach each node from the root.

The results show that the process chooses 1,579 genes, of which 299 correspond to up-regulated genes of the 345 that exist in the total data set, i.e. 86.7% of the total number of up-regulated genes. From the practical standpoint for the biological field, only 19% effectiveness has been achieved because there are 1,280 genes that are not up-regulated, which must be discarded using biological knowledge or by means of individual laboratory tests.

An alternative method for comparing these results is to use SOMs with the same data and conditions of the application with IENN. For this purpose the methodology proposed by Tamayo et al. [4] was followed, which suggests using SOMs in a single iteration, where the initial SOM mesh is fitted in such a way that at each node the patterns that present an increasing activity are identified. In this case the process shows that with a mesh of size 5x6 (30 nodes) it was possible to obtain patterns of increasing activity on the nodes of the SOM. The selected clusters are obtained directly from the patterns with increasing activity. With the SOMs 1,653 increasing activity genes were selected, 225 of which were up-regulated genes, and therefore in this case 65.2% of the 345 up-regulated genes were detected, and a practical efficiency of 13.6% was achieved, because 1,428 genes that do not correspond to up-regulated genes must be discarded.

Since in this application to the genes of E.Coli we can count on the 345 up-regulated genes [9] identified in the laboratory, it is possible to carry out an evaluation considering both methods (IENN and SOM) as classifiers. Moreover, if the expression level  $\theta$  is considered as a classification parameter, it is possible to make an analysis by means of Receiver Operating Characteristic (ROC), varying the expression level  $\theta$  over an interval of [4.4 - 8.9]. Figure 3 shows the ROC curves for IENN and SOM.

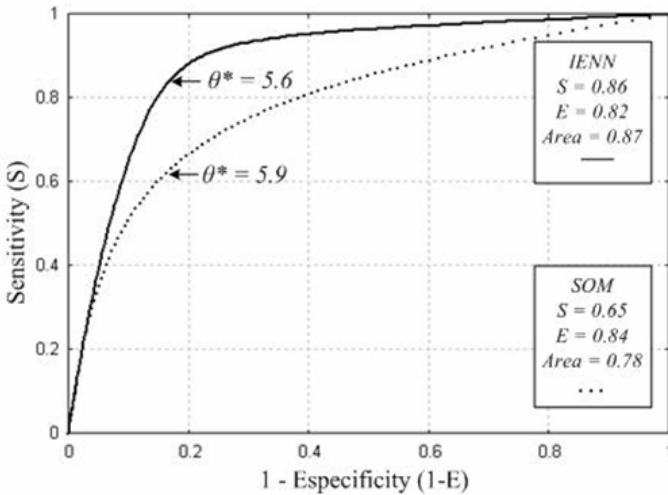


Fig. 3. ROC curves for IENN and SOM

The optimum classification value for IENN is achieved at  $\theta^*=5.6$ . At this point a sensitivity of 86% and a specificity of 82% were reached, covering an area of 0.87 under the ROC curve. When the same data, normalization values and expression level ranges were considered for SOM, an optimum classification value of  $\theta^*=5.9$  is obtained, achieving a sensitivity of 65%, a specificity of 84%, and an area under the ROC curve of 0.78.

## 4 Discussion and Conclusion

When the results of the proposed method (which uses ENN) are compared with those of the traditional SOM method, it is seen that the IENN method detects 74 up-regulated genes more than the SOM, which correspond to 21.5% of those genes. For practical purposes it must be considered that these genes are not recoverable in the case of the SOM because they are mixed up with the group of 5,659 undetected genes. On the other hand, the efficiency of the method that uses the ENN is better, because it requires discarding 1,280 genes that are not expressed, compared to the 1,428 that must be discarded with the SOM. Since the final objective of the experiment with E.Coli consists in detecting the up-regulated genes, it is possible to consider the IENN and SOM methods as classifiers and carry out an analysis of the merit of the classification by means of an ROC curve.

When considering an overall analysis of the classifier using the expression level  $\theta$  as a parameter, it is important to consider the area under the ROC curve. In this case the area for the proposed method is 0.87, compared to 0.78 for the SOM, which represents an 11.5% improvement. In relation to the sensitivity at the optimum decision level, the proposed method is 21% more sensitive than the SOM.

The numerical advantages derived from the application of the proposed method for the detection of the up-regulated genes of E.Coli are clear, but there are other aspects that must be analyzed with the purpose of projecting these results to the search of genes expressed in microarrays. The IENNs present several advantages that allow reinforcing the proposed method of iteration divisions. On the one hand, the IENNs have a greater capacity for adapting the net to the data because they have a set of parameters that control the deformation and elasticity properties. By carrying out the minimization of the overall energy in stages (evaluating different combinations of parameters  $\lambda$  and  $\mu$ ), a process similar to annealing is induced, making it possible to approach the overall minimum and not be trapped in local minimums. The same minimization methods allow the automatic selection of parameters that are fundamental for the later development of the process, such as the minimum expression level  $\theta_{\min}$  and the size of the net.

The other important advantage of the ENNs refers to their representation capacity, because the use of piecewise linear projecting makes it possible to increase the resolution of the data projected on the space having the lowest dimensions (internal coordinates). In the case of the microarray analysis this better representation becomes more important, since a common way of working in the field of microbiology and genetics is based on the direct observation of the data. On the other hand, the SOMs only allow a projection on the nodes when using *piecewise constant projecting* or the alternative U-matrix projections [2], [3], [4], which approximate only sets of data to the plane but do not represent directly each data.

A valid point that should be analyzed when comparing SOMs with IENNs is to consider the argument that an iteration process of divisions with SOMs can improve the results of the method. But the iteration process presented is based on the automatic selection of parameters (particularly the size of the net and the minimum expression level) for its later development, which is achieved by a global optimization method like ENN. The SOM does not allow the expression level to be determined automatically, and that information must come from the biological knowledge of the

expression levels of particular genes. The alternatives of using the minimum error of vector quantization of SOM as an alternative the minimum energy of ENN did not produce satisfactory results.

The results of the application to the discovery of up-regulated genes of E.Coli show a clear advantage of the proposal over the traditional use of the SOM method.

We chose to carry out a comparison with well established methods that are used frequently in the field of bioinformatics, but it is also necessary to evaluate other more recent alternatives such as flexible SOMs [13].

## References

1. Molla M, Waddell M, Page D and Shavlik J. Using machine learning to design and interpret gene-expression microarrays. *Artificial Intelligence Magazine* 25 (2004) 23-44.
2. Kohonen T. *Self-organizing maps*, Berlin: Springer-Verlag (2001).
3. Hautaniemi S, Yli-Harja O, Astola J. Analysis and visualization of gene expression microarray data in human cancer using self-organizing maps, *Machine Learning* **52** (2003) 45-66.
4. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander E and Golub T. Interpreting patterns of expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Genetics* **96** (1999) 2907-12.
5. Monti S, Tamayo P, Mesirov and Golub T. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Springer Netherlands*, Vol. **52**, (2003) 91-118.
6. Gorban A, and Zinovyev A. Method of elastic maps and its applications in data visualization and data modeling. *International Journal of Computing Anticipatory Systems, CHAOS*. **12** (2001) 353-69.
7. Gorban A, Zinovyev A, Wunsch D. Application of the method of elastic maps. In analysis of genetic texts. *Proc. International Joint Conference on Neural Networks (IJCNN)*, Portland, Oregon (2003) July 20-24.
8. Zinovyev AY, Gorban A, and Popova T, Self-organizing approach for automated gene identification, *Open Sys and Information Dyn* **10** (2003) 321-33.
9. Liu M, Durfee T, Cabrera T, Zhao K, Jin D, and Blattner F Global transcriptional programs reveal a carbon source foraging strategy by *E. Coli*. *J Biol Chem* **280** (2005) 15921-7.
10. Gorban A and Zinovyev A. Elastic principal graphs and manifolds and their practical applications, *Computing* **75** (2005) Springer-Verlag 359-379.
11. Quackenbush J, *Microarrays data normalization and transformation*, *Nature Reviews Genetics* **2** (2001) 418-27.
12. Maulik U, Bandyopadhyay S Performance evaluation of some clustering algorithms and validity indices, *IEEE PAMI* **24** (2002) 1650-4.
13. Salas R, Allende H, Moreno S and Saavedra C Flexible architecture of self-organizing maps for changing environments, *CIARP 2005, LNCS* **3773**, (2005) 642-53.

# A New Feature Selection Method for Improving the Precision of Diagnosing Abnormal Protein Sequences by Support Vector Machine and Vectorization Method

Eun-Mi Kim<sup>1</sup>, Jong-Cheol Jeong<sup>2</sup>, Ho-Young Pae<sup>3</sup>, and Bae-Ho Lee<sup>4</sup>

<sup>1</sup> Dept. of Computer Engineering,  
Chonnam National University, Republic of Korea  
koreaeunmi@yahoo.com

<sup>2</sup> Dept. of Electrical Engineering & Computer Science,  
The University of Kansas, USA  
jcjeong@ku.edu

<sup>3</sup> Dept. of Computer Engineering, Chonnam National University,  
Republic of Korea  
saint97@nate.com

<sup>4</sup> Dept. of Computer Engineering, Chonnam National University,  
Republic of Korea  
bhlee@chonnam.ac.kr

**Abstract.** Pattern recognition and classification problems are most popular issue in machine learning, and it seem that they meet their second golden age with bioinformatics. However, the dataset of bioinformatics has several distinctive characteristics compared to the data set in classical pattern recognition and classification research area. One of the most difficulties using this theory in bioinformatics is that raw data of DNA or protein sequences cannot be directly used as input data for machine learning because every sequence has different length of its own code sequences. Therefore, this paper introduces one of the methods to overcome this difficulty, and also argues that the capability of generalization in this method is very poor as showing simple experiments. Finally, this paper suggests different approach to select the fixed number of effective features by using Support Vector Machine, and noise whitening method. This paper also defines the criteria of this suggested method and shows that this method improves the precision of diagnosing abnormal protein sequences with experiment of classifying ovarian cancer data set.

## 1 Introduction

Pattern recognition and classification are essential technique for homology detection in bioinformatics. The reason that these issues can get interest in this area is that most data in bioinformatics exists based on gene sequences, and the sequences are different from one and the other. This property makes gene sequences are hard to be directly applied into a classifier. To overcome these

problems, many algorithms have been introduced, such as the Smith-Waterman dynamic programming algorithm, BLAST, FASTA, Profile, and Hidden Markov Models(HMM). However, some of them have very low efficiency although they require high computational complexity. Compared with these methods, recently introduced vectorization method has some advantages[7]. First, the pairwise score representation includes the concept of the profile HMM topology and parameterization; nevertheless, vectorization method is simpler than pervious algorithms Second, this method uses pairwise alignment, so compare with multiple alignment, the computational complexity are surprisingly decreased. Last advantage is that this method does not necessarily work on same domain like profile alignment does[7]. Although this method has several advantages on its theoretical background, and the results are surprisingly better than others, this method does not clearly mentioned about selecting features. This method assumed that there are several or finite number of well characterized representative sequences for producing scoring matrix based on pairwise alignment. The problem of this assumption is that how the well characterized can be selected? If there is the solution for selecting well characterized sequences, then why the solution cannot be directly used in the classification problem? If the well characterized sequences exist, then what does training data mean? Why the well selected sequences cannot be used as a portion of training data? Do the well characterized sequences have to be wasted? These several questions and the answers show that the purpose and object of this paper; therefore, this paper shows new approach that entire instances can be used as features and suggests the method and criteria for selecting features by reorganizing instances. Finally, this paper proves that the new suggested method improves the classification performance of vectorization methods, and also helps to prevent over-fitting problem. For the experiment, two kinds of Support vector machine are used as classifiers. One is using QP programming, and the other one is using neural networks for solving problems.

## 2 Modified SVM

The original SVM[3,4,8,11,12] is defined as followed by

$$\begin{aligned}
 L(\alpha) &= b \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j \\
 \text{subject to} \quad (1) \quad &\sum_{i=1}^N \alpha_i d_i = 0 \quad (2) \quad 0 \leq \alpha_i \leq C
 \end{aligned} \tag{1}$$

Where b is margin,  $\alpha$  is weighting vector, d is the destination of training data, which is used to expressed with positive class and negative class, and  $\mathbf{x}$  is input vector.

The modified SVM[4,12] is defined as followed by

$$\begin{aligned}
 L(\alpha) &= b \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{y}_i^T \mathbf{y}_j \\
 \text{subject to} \quad &0 \leq \alpha_i \leq C
 \end{aligned} \tag{2}$$

As shown above the equation is almost same except above equation changed original vector  $\mathbf{x}$  into augmented input vector  $\mathbf{y}$ , which is composed by input vector and the bias in the RBF neural networks and the second restriction in (1) has been omitted. The reason to augment input pattern is to simultaneously learn both bias and weight vector. This is very important because the omitted equation condition makes possible that SVM can be applied to sequential learning methods by substituting kernel  $K$  [3,4,12] for inner product pattern  $\mathbf{y}_i^T \mathbf{y}_j$ .

### 3 Relaxation

The modified SVM is applied into relaxation which is traditional classification in neural networks [13,16,12]. The procedure of traditional relaxation is shown below. To apply this method with SVM, it is necessary to change the input pattern,  $\mathbf{Y}$ , into linear or nonlinear kernel matrix. In our experiment, we used Radial Basis Function (RBF) Kernel. This method using kernel relaxation has several advantages and the one of them is that it is independent of memory problem and the processing time is huge improved. However, the problem of this learning algorithm is also inherited into the dual space, and this is the reason why the whitening method has been applied into the modified SVM because this algorithm has high possibility to face the over-fitting problem.

```

Given Training Data  $\mathbf{Y}, \mathbf{d}$ 
Begin initialize  $\mathbf{a} = \mathbf{0}, \eta(\bullet), m \arg \text{in } b > 0, k = 0$ 
do
    shuffling the training data
    if  $d_k \mathbf{a}^t \mathbf{y}_k \leq b$ 
        then  $\mathbf{a}(k+1) = \mathbf{a}(k) + \eta(k) \frac{d_k b - \mathbf{a}^t \mathbf{y}_k}{\|\mathbf{y}_k\|^2} \mathbf{y}_k$ 
    until  $d_k \mathbf{a}^t \mathbf{y}_k > b$  for all  $k$ 
return  $\mathbf{a}$ 
End
    
```

(3)

### 4 Dynamic Momentum

The basic concept of dynamic momentum scheduling is that the size of momentum value is getting decreased from initial state to convergence state. To apply this concept in the algorithm, the momentum has to be satisfied with certain conditions that the scale of the value of the momentum cannot exceed initialized momentum value, and the momentum value has to be regulated with smaller value than initial value [4,6,12].

$$\begin{aligned}
 \mathbf{M} &= \frac{m(k+1)}{\tau} \\
 \text{if } \mathbf{M} &> m \\
 \text{then } \mathbf{M} &= 0; \quad \tau = \tau^2;
 \end{aligned}$$
(4)

In (4), dynamic momentum  $\mathbf{M}$  is automatically initialized, and reorganizes next epoch tolerance  $\tau$  as changing its value to  $\tau^2$  when the value  $\mathbf{M}$  is bigger than the upper bound  $m$ ; therefore,  $\mathbf{M}$  is continuously learning its momentum value under given conditions. In conclusion, compared with existing static momentum scheduling which choose its momentum value by user externally, and it cannot be changed during its learning time, dynamic momentum can find momentum value actively as to be affected by learning epoch into given scheduling method [4,6,8,9,12].

### 5 Fisher’s Liner Discriminant

To approximate the pre-decision boundary, Fisher’s linear discriminant has been used. The simple review of the Fisher’s linear discriminant is shown below [13]. Input vector  $\mathbf{x}$ , the average of the sample  $\mathbf{m}$ , and within cluster scatter matrix  $\mathbf{S}_w$  are following by

$$\begin{aligned}
 \mathbf{m} &= \frac{1}{N_i} \sum \mathbf{x} \\
 \mathbf{m}_i &= \frac{1}{N_i} \sum_{\mathbf{x} \in D_i} \mathbf{x} \quad i = 1, 2 \\
 \mathbf{S}_W &= \sum_{i=1}^2 \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t
 \end{aligned}
 \tag{5}$$

$$\begin{aligned}
 \mathbf{w} &= \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \\
 w_0 &= -\mathbf{m}^t \mathbf{w}
 \end{aligned}
 \tag{6}$$

Therefore, Fisher’s linear discriminant can find the optimal direction for classifying the given data set by reducing multi dimension into one dimension.

**Table 1.** Vectorization method

Features \ Instances	S(i)	S(i+1)	...	S(k)
S(1)	Pw(S(1),S(i))	Pw(S(1),S(i+1))	...	Pw(S(1),S(k))
S(2)	Pw(S(2),S(i))	Pw(S(2),S(i+1))	...	Pw(S(2),S(k))
...	...	...	...	...
S(n)	Pw(S(n),S(i))	Pw(S(n),S(i+1))	...	Pw(S(n),S(k))

### 6 Vectorization

The vectorization method is pretty simple, and this is based on Smith-Waterman dynamic programming algorithm [7]. Let’s consider that there are a limit number of instances and features where both instances and features appears as gene sequences, and features are some what well characterized. From this,



Smith-Waterman dynamic programming algorithm performs as pairwise alignment between each instance and feature, and then this processing results score matrix which will be applied into classifiers. Vectorization method is shown below where Pw indicates Smith-Waterman dynamic programming algorithm.

## 7 The Noise Whitening Method as Feature Selection

The basic concept of the whitening method is that both preventing over-fitting problem and improving performance can be achieved by removing the noise which is abnormally located in the distribution of dataset. To explain this method, Gaussian distribution of two class data sets is assumed, and it is often true in practical dataset although they do not exactly follow the Gaussian distribution but it is likely to be true and this is enough to explain this method. Let's assume below situation that original data sets are extremely overlapped by some of data are located in the outside of the mid-point in opposite class. For convenience of explain, one class is called A, and the other one is called B. M1 is the center of the class A, and M2 is the center of class B. C1 is the mid-point between class A and B. As previously explained, the main concept of this method is to reduce noise data which is abnormally distributed, so here in whitening method assumes that the data which is located in the boundary between M1 and at most left hand side of data in class B, and M2 and at most right hand side of data in class A. In addition, the data a which is located between at most left hand side of class A and at most left hand side of class A', and the data b which is located between at most right hand side of class B and at most right hand side of class B' will be removed because these data do not affect to find decision boundary; therefore, we are finally dealing with the data set which is distributed between A' and B'. For the new feature selection, the data distributed in A' and B' are thresholded by a certain number. In this case, this paper proves that the number of maximum features cannot be more than 10 features. This is the criteria of this whitening feature selection method, and ironically, this method is based on reducing or removing abnormally distributed data, but it turns out that the results of this method are reducing features and

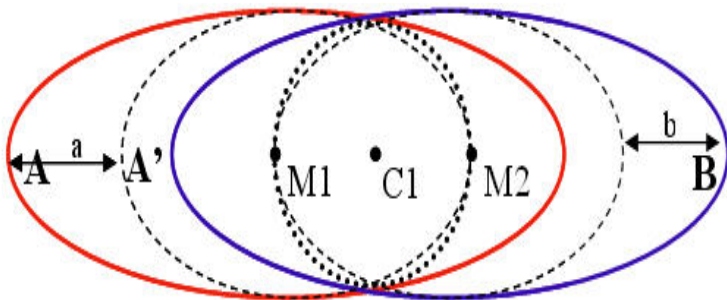


Fig. 1. The concept of noise whitening

selecting well characterized features. In addition, some of data between M1 and M2 also will be removed by sparse representation of SVM and this may help to increase the classification performance.

**Algorithm 1: noise whitening algorithm**

```

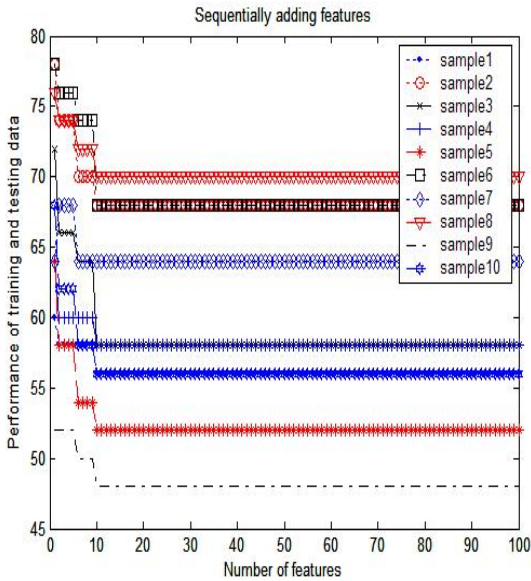
Given training data  $\mathbf{X}$ (n by m),  $\mathbf{y}$ (n by 1)
  Begin
    do
      Clustering training data using target information d
      Finding approximate center of a class using Fisher's method
      {
        Calculating mean vector for each class
        for  $i = 1$  to  $n$ 
          Calculating distance  $\varpi =$  between  $M1$  and  $X(i, :)$ 
           $\varpi = \bigcup \varpi_i; y = \bigcup y_i;$ 
        end
      }
    end
    Sorting  $\varpi$ 
     $\mathbf{X} =$  Reduced matrix which has been removed all other features
      except threshold features which is related with  $\varpi$ 
    Applying  $\mathbf{X}$  and  $\mathbf{y}$  into SVM
  end

```

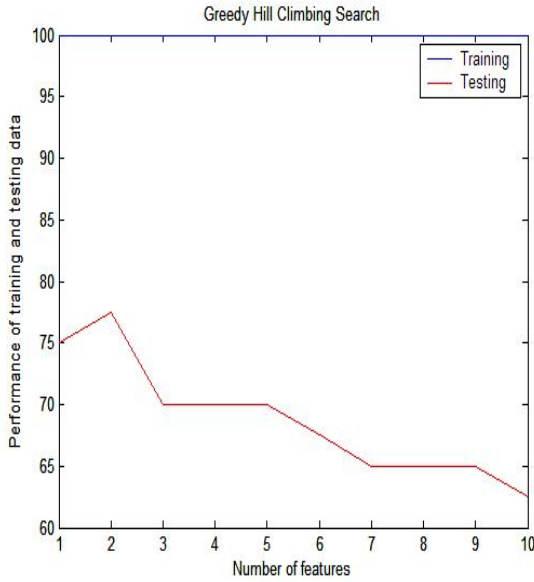
## 8 Experiment Results

Experiment data is composed of two class gene sequences. One sequence is human's protein sequence which got ovarian cancer, and the other sequence is normal sequence. This data is selected from National Center for Biotechnology Information. The number of instances is 100 where 50 are for ovarian cancer, and rest of them are normal. First 10 instances are assumed as well characterized sequences like previous vectorization method did. For this experiment we used two different scoring matrixes, BLOSUM62. First experiment is to approximate its performance with the number of features; therefore, in this case, both 100 features and 100 instances are applied into SVM1 classifier which used QP programming as previously mentioned, and then features are randomly selected with sequentially increasing the number of features. This performed 10 times, so this means that 10 differently ordered data sets are evaluated by sequentially increasing the number of features. Fig2 shows the result and the results are very interesting. All 10 different ways of sampling features have different performance, and this is pretty normal because each time, the processing selects different features, so it can result different performance and this is the reason why many

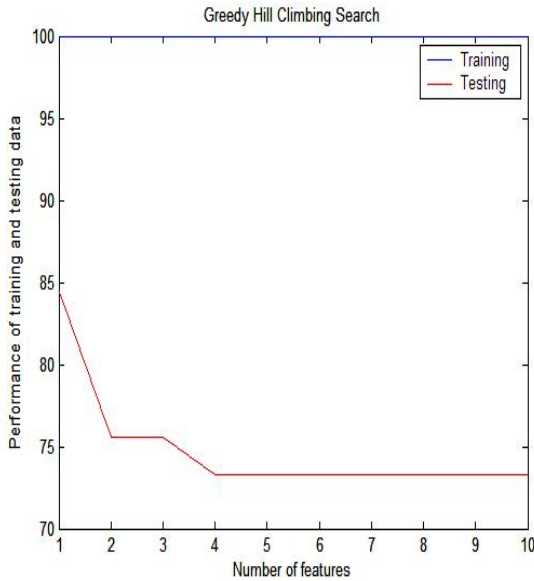
cases in bioinformatics need feature selection or cross validation methods. However, interestingly, in every case, the performance of classification is conversed after 10 features are added. This can be explained by the formation of score matrix which has diagonally distinctive values. This is obvious because the value of the score matrix is based on pair wise alignment, so the diagonal value will be most distinctive value among features because the score is of a perfect score; the score is from using same sequence for pair wise alignment. Therefore, the classification performance is strongly affected by diagonal, and this property causes that after certain number of features are added, the classification performance is soon converged. Mathematical proof is remained as further research issue. This experiment shows that the maximum number of necessary feature is 10% of the given instance. This is necessary to guarantee to get maximum performance of the given classification problem under vectorization method. Next experiment is to compare maximum performance between pre-assumed well characterized features and new feature selection method based on noise whitening. Fig3 shows that the testing performance of the well characterized feature appears between 77% and 63%. Fig4 appears that the performance of the new feature selection method by noise whitening, and the performances are located between 84% and 74%. This results are based on greedy hill climbing search; therefore, this experiment shows that the potential classification performance of the given features. The resulted potential performances show that new feature selection method outperformed previous vectorization method.



**Fig. 2.** The performance of classification with randomly sampled 10 different data set and sequentially adding features



**Fig. 3.** The classification performance of pre-assigned well characterized features



**Fig. 4.** The classification performance of new feature selection method using noise whitening

## 9 Conclusions

This paper suggests criteria that the previous vectorization needs at least 10% features of the number of instances to get maximized performance, and more than 10% of the number of instances can decrease its potential performance. The suggested feature selection method is based on removing noise data and improving purity of the given data set although this concept has defect that it can be waste data. However, this method cannot consider this shortcoming because in the vectorization method the instances are considered exactly same as features, so reducing dataset can be considered as selecting features. Finally, the results of the experiment show that the suggested feature selection method outperforms the previous vectorization method. The mathematical proof of 10% criteria is remained as further study.

## References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, Inc. New York. NY. (2001)
2. Hansen, P.C.: Regularization Tools, A Matlab Package for Analysis and Solution of Discrete Ill-Posed Problems. Version 3.1 for Matlab 6.0, (2001)
3. Haykin, S.: Neural Networks, A comprehensive Foundation. Prentice-Hall Inc. (1999)
4. Jeong, J.C.: A New Learning Methodology for Support Vector Machine and Regularization RBF Neural Networks. Thesis for the degree of the master of engineering. Department of Computer Engineering Graduate School. Yosu National University. Republic of Korea. (2002)
5. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of ECML-98, 10th European Conference on Machine Learning. (1998)
6. Kim, E.M, Park, S.M, Kim, K.H., Lee, B.H.: An effective machine learning algorithm using momentum scheduling. Hybrid Intelligent Systems. Japan. (2004) 442–443
7. Li, L., William. S. N.: Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationship. Journal of Computational Biology. **10**(6). (2003) 857–867
8. Mangasarian O.L., Musicant, D.R.: Active Set Support Vector Machine Classification. Neural Information Processing Systems 2000 (NIPS 2000). T. K. Lee, Dietterich, T. G., Tresp, V. editors. MIT Press. (2001). 577–583
9. Platt, J.C.: Fast Training of Support Vector Machines Using Sequential Minimal Optimization, In Advances in Kernel Methods: Support Vector Learning. MIT Press. Cambridge (1998)
10. Tikhonov, A.N.: On solving incorrectly posed problems and method of regularization. Doklady Akademii Nauk USSR. **151** (1963) 501–504
11. Vapnik, V.: Statistical learning theory. John Wiley and Sons. New York (1998)
12. Yoo, J.H., Jeong, J.C.: Sparse Representation Learning of Kernel Space Using the Kernel Relaxation Procedure. Journal of Fuzzy Logic and Intelligent Systems. **11**(9) (2002) 817–821

# Epileptic Seizure Prediction Using Lyapunov Exponents and Support Vector Machine

Bartosz Świdorski<sup>1</sup>, Stanisław Osowski<sup>1,2</sup>, Andrzej Cichocki<sup>1,3</sup>, and Andrzej Rysz<sup>4</sup>

<sup>1</sup> Warsaw University of Technology, Warsaw, Poland  
Koszykowa 75, 00-662 Warsaw, Poland  
{sto, markiewt}@iem.pw.edu.pl

<sup>2</sup> Military University of Technology, Warsaw, Poland

<sup>3</sup> Brain Science Institute, RIKEN, Japan

<sup>4</sup> Banach Hospital, Warsaw, Poland

**Abstract.** The paper presents the method of predicting the epileptic seizure on the basis of EEG waveform analysis. The Support Vector Machine and the largest Lyapunov exponent characterization of EEG segments are employed to predict the incoming seizure. The results of numerical experiments will be presented and discussed.

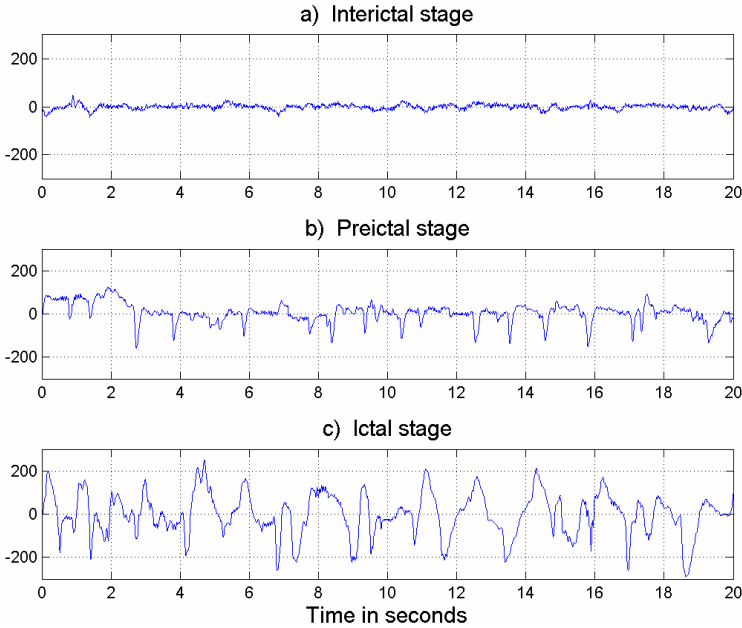
## 1 Introduction

Epilepsy is a group of brain disorders characterized by the recurrent paroxysmal electrical discharges of the cerebral cortex, that result in irregular disturbances of the brain functions [3]. The spatio-temporal dynamical changes in EEG, beginning several minutes before and ending several minutes after the seizure, evolve in a characteristic pattern, culminating in a seizure.

It has been shown [2] that EEGs do not merely reflect stochastic processes, and instead that they manifest deterministic chaos. Using the EEG recording of a human epileptic seizure the papers [1], [5], [6] have shown the existence of a chaotic attractor, being the consequence of the deterministic nature of the brain activity.

In this paper we will investigate the phenomena of the EEG internal dynamics, characteristic for the epileptic activity, through the use of the short-term largest Lyapunov exponent (STLmax) and Support Vector Machine (SVM). The Lyapunov exponents and the other measures based on them will serve as the features used by the SVM to predict the incoming seizure. This approach is motivated by the special properties observed in EEG waveforms in the baseline (interictal) and epileptic states.

The typical EEG waveforms corresponding to three stages of an epileptic seizure: interictal, preictal and ictal are shown in Fig. 1. The ictal stage corresponds to the seizure period. The preictal stage is the time period preceding directly the seizure onset. No strict borders of the preictal period are defined. The postictal stage follows the seizure end. The interictal stage is the period between the postictal stage of one seizure and the preictal stage of the next seizure. In this state the EEG signal is chaotic of very unpredictable nature and relatively small magnitude. Observing the



**Fig. 1.** The typical EEG waveforms corresponding to epilepsy: a) interictal, b) preictal and c) ictal stage (*horizontal axis* - seconds, *vertical axis* – microvolts)

EEG signals in all stages we can understand the difficulties in recognizing the seizure onset. The differences between the preictal and ictal stages are hardly noticeable for non-specialist.

As we approach the epileptic seizure the signals are less chaotic and take more regular shape. Moreover observing many channels at the same time we can see that the signals of all channels are becoming coupled and to some degree locked. The simultaneous synchronization of signals of many channels proves the decreasing chaoticity of signals. Thus the chaoticity measures of the actual EEG signal may provide good features for the prediction of the incoming seizure.

## 2 Largest Lyapunov Exponents for Characterization of EEG

Each EEG signal recorded from any site of the brain may be associated with the so called embedding phase space. If we have a data segment  $x(t)$  of duration  $T$  we can define the vector  $\mathbf{x}_i$  in the phase space as following [5]

$$\mathbf{x}_i = [x(t_i), x(t_i + \tau), \dots, x(t_i + (p-1)\tau)] \quad (1)$$

with  $\tau$  - the selected time lag between the components of each vector,  $p$  - the selected dimension of the embedding space and  $t_i$  - the time instant within the considered period  $[T - (p-1)\tau]$ . Each vector  $\mathbf{x}_i$  in the phase space represents an instantaneous state of the system.

The Lyapunov exponents have been proven to be useful dynamic diagnostic measure for EEG waveforms [1], [2], [4], [10]. Lyapunov exponents define the average exponential rate of the divergence or convergence of the nearby orbits in the phase space. It may be described in the form  $d(t) = d_0 e^{Lt}$ , where  $L$  means the Lyapunov exponent and  $d_0$  – the initial distance of two nearby orbits. The magnitude of the exponent reflects the time scale on which the system dynamics become unpredictable [1],[5]. Any system containing at least one positive Lyapunov exponent is defined to be chaotic and the nearby points, no matter how close, will diverge to any arbitrary separation.

The estimation  $L$  of the  $STL_{\max}$  exponent may be presented as following [1], [4]

$$L = \frac{1}{N\Delta t} \sum_{i=1}^N \log_2 \frac{|\Delta \mathbf{x}_{ij}(\Delta t)|}{|\Delta \mathbf{x}_{ij}(0)|}, \quad (2)$$

where  $\Delta \mathbf{x}_{ij}(0) = \mathbf{x}(t_i) - \mathbf{x}(t_j)$  is the displacement vector at the time point  $t_i$ , that is the perturbation of the fiducial orbit observed at  $t_j$  with respect to  $t_i$ , while  $\Delta \mathbf{x}_{ij}(\Delta t) = \mathbf{x}(t_i + \Delta t) - \mathbf{x}(t_j + \Delta t)$  is the same vector after time  $\Delta t$ . The vector  $\mathbf{x}(t_i)$  is the point in the fiducial trajectory for  $t = t_i$  and  $\mathbf{x}(t_j)$  is a properly chosen vector adjacent to  $\mathbf{x}(t_i)$  in the phase space. The time increase  $\Delta t$  is the evolution time, that is the time which is allowed for  $\Delta \mathbf{x}_{ij}$  to evolve in the phase space. For the time given in sec the value of  $L$  is in bits/sec.  $N$  is the number of local  $STL_{\max}$ 's that will be estimated within period  $T$  of the data segment, where  $T = N\Delta t + (p-1)\tau$ . For the exponent  $L$  to be a good estimate of  $STL_{\max}$  the candidate vector  $\mathbf{x}(t_j)$  should be chosen in such a way that the previously evolved displacement vector  $\Delta \mathbf{x}_{i-1,j}(\Delta t)$  is almost parallel to the candidate displacement vector  $\Delta \mathbf{x}_{ij}(0)$ . Moreover  $\Delta \mathbf{x}_{ij}(0)$  should be small in magnitude to avoid computer overflow in the evolution within chaotic region.

In this work we have applied special method of choosing two time segments of EEG taking part in estimation of the largest exponent [12]. It is based on the statistical hypothesis that two chosen time segments are alike to each other. Each time segment is characterized by the vector  $\mathbf{x}$  described by (1) of the length equal  $p=8$ . The length of the vector was chosen in such a way that the time series within this time span may be regarded as a stationary and at the same time the series is long enough to provide stabilization of the estimate to obtain the credible results of the performed calculations. We have applied the Kolmogorov-Smirnov test [7]. Let us denote the set of points belonging to the observed fiducial EEG trajectory corresponding to time  $t_i$  by  $\mathbf{x}_i$  as shown by equation (1). By  $\mathbf{x}_j$  we denote the vector generated from the same EEG waveform placed at  $t_j$  not far from the time point  $t_i$ . Similarity of these two vectors in the Kolmogorov-Smirnov test is characterized by the minimal significance level  $h$  needed to reject the null-hypothesis  $H_0$  that these vectors are drawn from the same underlying continuous population, that is belong to the same distribution (at the assumed dimension  $p$  of the vectors). Let  $H_1$  be the alternative hypothesis. On the basis of the observed actual statistics of the Kolmogorov-Smirnov test we determine the minimal significance level  $h$  needed to reject the null hypothesis. High value of the actual significance level  $h$  means high similarity of the processes, characterized by the vectors  $\mathbf{x}_i$  and  $\mathbf{x}_j$  (small distance between both vectors). For the highest possible value of  $h$  they contribute to the summation terms in the equation (2).



For each data segment of the length  $T$  we choose the vectors  $\mathbf{x}_i$  ( $i=1, 2, 3\dots$ ) and each of these vectors is compared with many vectors  $\mathbf{x}_j$  formed from the data belonging to the same segment. The vectors  $\mathbf{x}_j$  are generated at the initial points  $t_j$  placed at different distances, starting from the distance larger than three times the length of the vector. For each  $\mathbf{x}_i$  many pairs  $(\mathbf{x}_i, \mathbf{x}_j)$  are compared and the significance levels  $h$  calculated. The vector  $\mathbf{x}_j$  corresponding to the highest value of  $h$  is selected and used together with  $\mathbf{x}_i$  in the process of the determination of largest Lyapunov exponent according to equation (2).

The value  $h$  of the significance level needed to reject the null hypothesis has been also used by us for the determination of the distance  $|\Delta\mathbf{x}_{ij}|$  between two vectors  $\mathbf{x}_i$  and  $\mathbf{x}_j$  that contribute to the Lyapunov exponent. We have applied here the measure of distance defined in the form

$$|\Delta\mathbf{x}_{ij}| = 1 - h. \tag{3}$$

This measure was used in the relation (2) for the determination of  $|\Delta\mathbf{x}_{ij}(0)|$  and  $|\Delta\mathbf{x}_{ij}(\Delta t)|$  in all our experiments.

The important observation from many experiments is that in the direct preictal stage we can observe the phenomena of progressive locking of the EEG waveforms observed at many sites of the brain. To quantify many observed sites simultaneously we have introduced some measure of similarity of profiles. We have employed the  $T_{index}$  from the well known t-test statistics as a measure of distance between the mean values of pairs of  $STL_{max}$  profiles over time [7]. The value of  $T_{index}$  at the time  $t$  between the electrode  $i$  and  $j$  is defined as

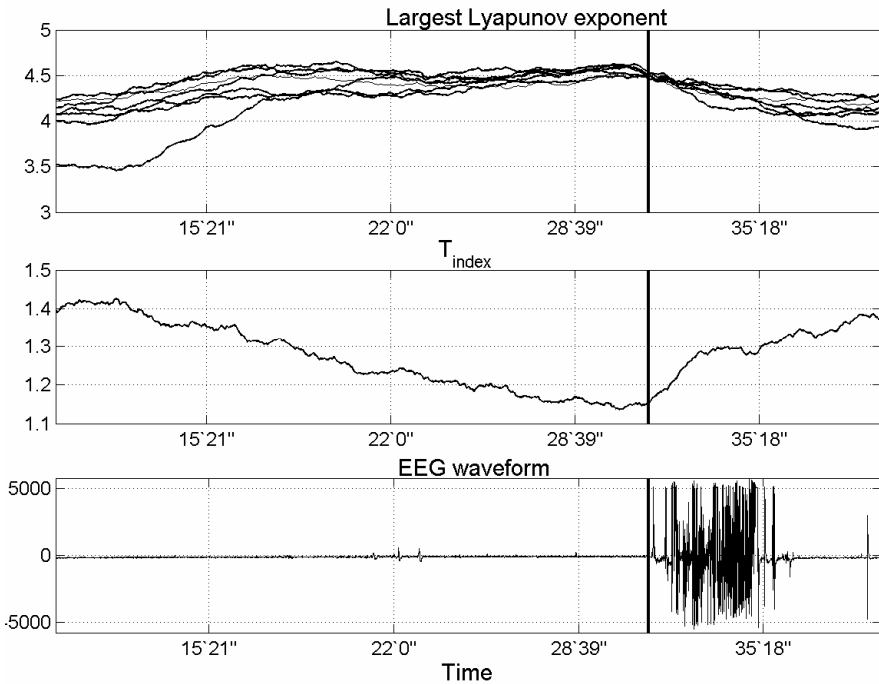
$$T_{ij}(t) = \frac{E\left\{ \left| STL_{max,i}(t) - STL_{max,j}(t) \right| \right\}}{\sigma_{i,j}(t) / \sqrt{N}}, \tag{4}$$

where  $E\{\}$  means the average of all absolute differences  $|STL_{max,i}(t) - STL_{max,j}(t)|$  within a moving window  $w_t$ , where  $w_t$  is equal 1 in the interval  $[t-N+1, t]$  and zero elsewhere, while  $N$  is the length of the moving window. The variable  $\sigma_{ij}(t)$  is the sample standard deviation of  $STL_{max}$  differences between electrode sites  $i$  and  $j$  within the moving window  $w_t$ . To observe the general trend for all electrodes at the same time we define the global  $T_{index}$  value for the combination of all pairs of electrodes

$$T_{index} = E\{T_{ij}\}. \tag{5}$$

The effect of gradual locking of EEG registrations at different channels is manifested in the form of the decreasing trend of  $T_{index}$  versus time. In the vicinity of the onset point the  $T_{index}$  assumes the minimum value.

Fig. 2 presents the typical EEG waveform (lower subplot) registered within the period of 40 min with the seizure occurring at 31min. The seizure point indicated by the neurologist is denoted by the vertical line. The upper and middle subplots present respectively, the changes of the largest Lyapunov exponents of the EEG of 8 channels and the  $T_{index}$  measure corresponding to these channels. The time axis is



**Fig. 2.** The typical change of the Lyapunov exponents of the epileptic EEG waveform

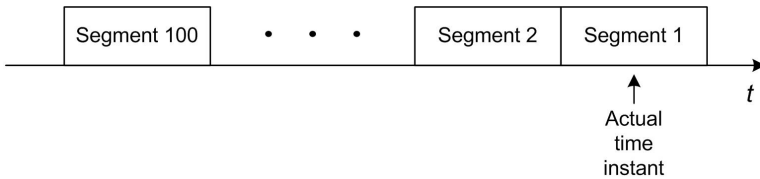
given in minutes and seconds. The seizure onset represents a temporal transition of the system from a chaotic state to a less chaotic one, corresponding to a synchronized rhythmic firing pattern of neurons participating in a seizure discharge. So it may be associated with the drop in the Lyapunov exponent values (the decreasing trend) determined for all electrode sites participating in the seizure. The synchronization of the channels is manifested by the decreasing trend of  $T_{\text{index}}$  measure.

### 3 Prediction of the Seizure Using Support Vector Machine

The main goal of the research is elaboration of the method for prediction of the incoming epileptic seizure on the basis of the registered EEG waveform. To solve the problem we have applied the Support Vector Machine working in the classification mode and EEG characterization using largest Lyapunov exponents. The SVM network is trained to recognize the time span corresponding to the preictal stage, starting 10 minutes before the potential seizure and lasting up to the seizure moment (destination  $d=1$ ). The time span outside this period (in the interictal stage) is associated with the other destination ( $d=-1$ ). So the SVM network works in a classification mode.

### 3.1 Feature Generation

The most important point in training the SVM classifier is the generation of the diagnostic features forming the vector  $\mathbf{x}$  applied to its inputs. We have made use of the observation that the seizure is strictly associated with some pattern of changes of the Lyapunov exponent preceding it. To take into account the change of the Lyapunov exponents within time we consider the EEG waveform split into 10 second segments. Fig. 3 shows the way the EEG waveform is split into 100 segments, each of approximately 10 seconds duration. For each segment we determine the Lyapunov exponent  $L$  and  $T_{index}$  associated with the channels under observation. On the basis of these values we generate the first set of diagnostic features used by the SVM network in the prediction process.



**Fig. 3.** The division of the EEG waveform into 100 segments

This set of features is obtained from the polynomial approximation of the series of 100 Lyapunov exponent  $L$  and  $T_{index}$  values. The values of the polynomial coefficients are generated for the actual segment and 99 past segments, directly preceding it. We apply the polynomial of the second order  $P(x)=ax^2+bx+c$ , built separately for Lyapunov exponent ( $x=L$ ) and for  $T_{index}$  ( $x=T_{index}$ ). The coefficients  $a$ ,  $b$  and  $c$  for both cases create the features (6 features together).

The next features are equal to the standard deviations of these 100 Lyapunov exponent and  $T_{index}$  values (2 features). The last set of features is generated from the ARMA model [7] of the EEG waveform of the actually considered segment (segment No 1). We have applied (1,3) ARMA model of four parameters forming the features. In this way the input vector  $\mathbf{x}$  to the SVM is composed of 12 components (6 features following from polynomial approximation, 2 standard deviation values and 4 coefficients of ARMA model). Observe that one input vector  $\mathbf{x}$  needs the analysis of the EEG waveform of duration of approximately 1000 seconds. The input data set has been generated for the segments moving each time by 10 seconds. In this way the EEG recording of few hours may result into few hundreds of data points. Some of them have been used in learning and the rest in the testing mode.

### 3.2 Support Vector Machine

The prediction task is performed in our system by the Support Vector Machine network (SVM) known as the excellent tool of good generalization ability [11]. The Gaussian kernel SVM based classifiers have been found the best. The principle of operation of such classifier and the review of the learning algorithms can be found for example in the book [11].

The important point in designing SVM classifier is the choice of the parameter  $\sigma$  of the Gaussian kernel function and the regularization parameter  $C$ . The parameter  $C$  controls the tradeoff between the complexity of the machine and the number of non-separable data points used in learning. The small value of  $C$  results in the acceptance of more not separated learning points. At higher value of  $C$  we get the lower number of classification errors of the learning data points, but more complex network structure. The optimal values of  $C$  and  $\sigma$  were determined after additional series of learning experiments through the use of the validation test sets. Many different values of  $C$  and  $\sigma$  combined together in the learning process have been used in the learning process and their optimal values are those for which the classification error on the validation data set was the smallest one.

The whole set of data has been split into the learning set (60%) and testing one (40%). The testing set is used only for testing the trained system. The number of representatives of the seizure and no seizure warning periods have been balanced as much as possible in both sets. The SVM network was trained using the learning data and the trained network tested on the testing data not used in learning. The hyperparameters  $C$  and  $\sigma$  have been adjusted by applying the validation data (20% of the learning data). The final values of these parameters used in experiments were  $C=500$  and  $\sigma = 0.8$ .

## 4 The Results of the Numerical Experiments

The numerical experiments have been performed for 15 patients of the Banach Hospital in Warsaw. Seven of them suffered from the temporal lobe epilepsy and eight from frontal lobe epilepsy. The EEG waveforms have been registered at 8 electrodes placed on the scalp with the sampling rate of 250 samples per second. They have been filtered by the low pass filter of the cut-off frequency 70Hz.

The estimations of the Lyapunov exponent values have been performed by dividing the recorded signal into the segments of  $T=10$  seconds. The embedding dimension  $p$  applied in the experiments was equal 8, while the other parameters:  $\tau = 0.016s$ ,  $\Delta t = 45s$ . The whole segment of the length  $T$  was divided into 270 subsegments and each of them was associated with the vector  $\mathbf{x}$  for the largest Lyapunov exponent estimation.

To assess the results we have defined some measures of the prediction quality. Let  $p$  be the number of data segments forming the testing data, from which  $p_s$  denote the number of segments indicating the incoming seizure and  $p_n$  the number of segments outside the seizure region ( $p = p_s + p_n$ ). By

$$\varepsilon = \frac{\Delta p}{p} \quad (6)$$

we denote the global misclassification rate, as the ratio of erroneous predictions ( $\Delta p$ ) to the total number ( $p$ ) of segments.

For the learning purposes we have assumed the period of 10 minutes of the preictal stage just before the seizure, for which the assumed destination was 1 (warning of the incoming seizure). The interictal period used in learning has covered the time span up to 20 minutes prior to seizure and these patterns have been associated with the

alternative destination (no seizure warning). The learning and testing data have been generated for each patient separately and the separate SVM networks have been trained for each patients (15 SVM networks). In the testing phase we have observed the real advance time  $T_p$  of prediction and the total testing error  $\varepsilon$ . Additionally for each patient we have noticed the total time of recorded EEG and the number of segments belonging to the preictal ( $p_s$ ) and interictal ( $p_n$ ) stages, used in experiments.

**Table 1.** The results of testing the SVM system for seizure prediction

Patient	Time of EEG recording <i>hour:min:sec</i>	$p_s$	$p_n$	$\varepsilon$ [%]	$T_p$ <i>min:sec</i>
1	1:55:49	137	134	16.54	9:06
2	1:56:34	134	138	0.73	10:04
3	0:58:28	137	134	11.39	10:08
4	0:58:33	138	134	1.84	10:12
5	0:54:24	136	136	14.34	9:45
6	0:30:43	70	72	17.6	14:25
7	1:38:10	140	132	10.3	10:07
8	0:58:57	136	136	8.83	8:54
9	1:47:00	134	136	8.89	9:52
10	1:41:03	136	136	9.19	9:40
11	2:01:36	139	133	4.04	10:12
12	1:21:39	137	135	12.13	9:35
13	0:42:08	136	136	16.19	9:57
14	2:01:14	132	138	1.85	10:05
15	0:52:25	133	139	6.98	10:13

Table 1 depicts the results of prediction of the seizure for 15 patients suffering from epilepsy. For each patient there was one seizure moment determined by the neurologist. All results depicted in the table correspond to the testing data, not taking part in learning. Observe that the prediction problem has been transformed to the classification task.

The average misclassification rate  $\varepsilon$  for 15 patients was equal 9.39% and the advance prediction time was in a very good agreement with the time assumed in learning phase (10 minutes). This confirms our conjecture of strict connection of the patterns of change of the largest Lyapunov exponents with the incoming seizure.

## 5 Conclusions

The paper has presented the application of the analysis of the EEG waveform and the SVM network to the prediction of the epileptic seizure of the brain. We have assumed the chaotic model of the brain activity and applied the largest Lyapunov exponent for its characterization. The employed measures do not assume any particular nonlinear model of the process.

The results of numerical simulations have confirmed that this model of prediction of the incoming seizure has great potentiality of application for individual patients. The obtained average accuracy of prediction was approximately 91% at the advance time span of prediction close to 10 minutes.

**Acknowledgements.** The paper has been supported by the Brain Science Institute, RIKEN, Japan.

## References

1. Babloyantz, A., Destexhe A.: Low dimensional chaos in an instance of epilepsy. *Proc. Natl. Acad. Sci, USA* 83 (1986) 3513-3517
2. Freeman W. J.: Strange attractors that govern mammalian brain dynamics shown by trajectories of EEG potentials. *IEEE Trans. CaS.* 35 (1988) 781-784
3. Gevins A., Remond A.: *Handbook of EEG and clinical neurophysiology.* (1987) Elsevier, Amsterdam
4. Iasemidis L. D., Principe J. C., Sackellares J. C.: Measurement and quantification of spatio-temporal dynamics of human seizures. (in "Nonlinear Signal Processing in Medicine", ed. M. Akay) IEEE Press, (1999) 1-27
5. Iasemidis L. D., Shiau D. S., Chaovalitwogse W., Sackellares J. C., Pardalos P. M., Principe J. C., Carney P. R., Prasad A., Veeramani B., Tsakalis K.: Adaptive epileptic seizure prediction system. *IEEE Trans. Biomed. Eng.* 50 (2003) 616-627
6. Iasemidis L., Sackellares J. C.: Chaos theory in epilepsy. *The NeuroScientist*, 2, (1996) 118-126
7. Matlab with toolboxes. (2002) MathWorks, Natick, USA
8. Nikias C., Petropulu A.: Higher order spectral analysis. (1993) Prentice Hall, N. J.
9. Osowski S.: *Neural networks for signal processing.* (2000) OWPW Warsaw
10. Palus M., Albrecht V., Dvorak I.: Information theoretic test for nonlinearity in time series. *Phys. Lett. A* 175, (1993) 203-209
11. Schölkopf B., Smola A.: *Learning with kernels.* (2002) Cambridge MA, MIT Press
12. Swiderski B., Osowski S., Rysz A.: Lyapunov Exponent of EEG Signal for Epileptic Seizure Characterization. (2005) IEEE Conf. European Circuit Theory and Design Cork

# Classification of Pathological and Normal Voice Based on Linear Discriminant Analysis

Ji-Yeoun Lee, SangBae Jeong, and Minsoo Hahn

Speech and Audio Information Lab.,  
Information and Communications University, 119, Munjiro, Yuseong-gu, Daejeon,  
305-732, Korea  
{jyle278, sangbae, mshahn}@icu.ac.kr

**Abstract.** This paper suggests a new method to improve the performance of the pathological/normal voice classification. The effectiveness of the mel frequency-based filter bank energies using the fisher discriminant ratio (FDR) is analyzed. Also, mel frequency cepstrum coefficients (MFCCs) and the feature vectors through the linear discriminant analysis (LDA) transformation of the filter bank energies (FBE) are implemented. In addition, we emphasize the relation between the pathological voice detection and the feature vectors through the FBE-LDA transformation. This paper shows that the FBE LDA-based GMM is a sufficiently distinct method for the pathological/normal voice classification. The proposed method shows better performance than the MFCC-based GMM with noticeable improvement.

## 1 Introduction

Many approaches to analyze the acoustic parameters for the objective judgment of the pathological voice have been developed. Among the acoustic parameters, the important parameters are pitch, jitter, shimmer, the harmonics to noise ratio (HNR), and the normalized noise energy (NNE). Good correlations between the parameters and pathological voice detection have been demonstrated and these parameters are based on the fundamental frequency[1-3]. However, it is not easy to correctly estimate the fundamental frequency in the pathological voice.

Many studies to analyze the short-time eletroglottographic (EGG) signal have been reported. One of the well-known methods is to detect pathological voices from the excitation waveform extracted by inverse filtering[4-6]. However, the linear predictive coding (LPC) based on inverse filtering needs the assumption of a linear model. It may not be the proper method because the speech pathology is mainly caused by the vocal source non-linearity.

In last years, pattern classification algorithms such as the Gaussian mixture model (GMM), the neural networks (NN), the vector quantization (VQ) and the characteristic parameter such as mel frequency cepstral coefficients (MFCCs) become more popular for the voice damage detection. Especially, the GMM and MFCCs become generally accepted as most useful methods for the detection of voice impairments as in [7][8][9].

The primary purpose of this paper is to develop an efficient method to detect the pathological voice and to improve the performance. First, our study examines the effectiveness of the mel frequency-based filter bank energies as fundamental parameters of the feature extraction using the fisher discriminant ratio (FDR). And then performance of MFCCs using the GMM for the construction of the baseline system is measured. Finally, a new approach with the LDA transformation of the FBE is suggested. Our experiments and analysis verify the effectiveness of the parametric space extracted from the FBE-LDA transformation compared to the MFCCs space.

This paper is organized as follows. Chapter 2 shows the previous works, the GMM and MFCCs, to detect the voice impairments. Chapter 3 suggests an effective method for the performance improvement, the FBE-LDA. Chapter 4 describes the overall procedure of our suggested method, analyzes the mel frequency-based filter bank energies and explains the experiments and improved results. Finally, chapter 5 is for conclusion.

## 2 Previous Works

The GMM as the pattern classification algorithm and MFCCs as the feature vectors are in common use. Recently, Godino-Llorente's research[9] shows the integral results of GMM and MFCCs. The cause of the pathological voice is an asymmetrical movement and an incomplete closure due to an increase of the vocal folds' mass. That is, the speech pathology happens to the problem in the vocal source. To extract the pathology-specific characteristics, it is necessary to separate out excitation component from the vocal tract. It can be implemented by the cepstral analysis. The speech signal  $s(t)$  is considered as the convolution of the excitation source  $x(t)$  and the vocal tract filter  $v(t)$  in time domain like (1). According to the convolution theorem, the corresponding signal in the frequency domain is equivalent to the spectral multiplication of the excitation source  $X(f)$  and the vocal tract filter  $V(f)$  components as in (2).

$$s(t) = x(t) * v(t) = \int_{n=-\infty}^{\infty} x(n)v(t-n) \quad (1)$$

$$S(f) = X(f) * V(f) \quad (2)$$

When the spectrum is logarithmically represented, the components become additive due to the property of the logarithm as  $\log(a * b) = \log(a) + \log(b)$ . Once the two components are additive, it can straightforwardly separate them using filtering technique. It is a basic principle of the MFCCs extraction and this is the reason that we use the MFCCs to obtain the excitation source. The detailed MFCCs extraction procedure is can be found in [7][8]. On the other hand, the reasons that the GMM is most frequently used than other algorithms are as follows: first, the GMM is able to model the set of the acoustic classes to express the phoneme such as a vowel, a nasal, etc. Second, the linear combination of the Gaussian basis function can express the characteristic distribution of the



speaker with mean vectors, covariance matrixes and mixture weights of each component's density. The procedure of the GMM can be found in [7][9].

### 3 Linear Discriminant Analysis

The LDA aims at finding the best combination of classes to improve the discrimination among the feature vector classes. It is implemented by the linear transformation matrix of the feature vector classes. The transformation is defined as the method to maximize  $tr(W^{-1}B)$ , where  $tr(M)$  denotes the trace of matrix  $M$ .  $W$  and  $B$  are the within and between class covariance matrices defined as [9][10]:

$$\begin{aligned}
 W &= \frac{1}{N} \sum_{k=1}^K \sum_{n=1}^{n_k} (x_{kn} - \mu_k)(x_{kn} - \mu_k)^t, \\
 B &= \frac{1}{N} \sum_{k=1}^K n_k (\mu_k - \mu)(\mu_k - \mu)^t
 \end{aligned} \tag{3}$$

$N$ : the total number of training patterns,  $K$ : the number of classes  
 $n_k$ : the number of training patterns of the  $k^{th}$  class,  
 $\mu_k$ : mean of the  $k^{th}$  class,  $\mu$ : overall mean

The transformation matrix is formed by the eigenvectors corresponding to the predominant eigenvalues, the largest eigenvalues of the matrix  $tr(W^{-1}B)$ , in the classes.

The LDA can be implemented in two forms: class-independent and class-dependent transformations. The class-independent method maximizes the ratio of the class covariances across all classes simultaneously. This defines the single transformation matrix in  $K$  classes. The class-dependent method implemented in this paper maximizes the ratio of the class covariances for each class separately. This forms the  $K$  transformation matrixes, each corresponding to one class.

## 4 Experiments and Results

### 4.1 Database

The disordered voice database distributed in the Kay Elemetrics is used to our experiments[11]. It includes 53 normal and 657 pathological speakers with the voice disorders. The acoustic samples are the sustained phonation of the vowel /ah/. In this paper, a subset is formed to 53 normal and 600 pathological speakers from the above database. To maintain the balance of the number of speakers, 547 normal Korean speakers' data are added. Each file is down-sampled to 16 kHz and quantized by 16 bits. 70% and 30% of our data are used for training and test, respectively.

### 4.2 Overall Block Diagram

Fig.1 (a) and (b) presents the training and test process in the overall block diagram of our pathological/normal voice classification procedure. Firstly, the mel frequency-based filter bank energies from the voice sample are estimated. They are the important baseline feature vectors utilized in our procedure. Their analyses are implemented by two methods of the feature extraction to compare the performances: (\*) MFCCs extraction through the discrete cosine transform (DCT) and (#) extraction of the feature vectors through the FBE-LDA transformation. In Fig.1 (a), Gaussian models of the pathological/normal voices are trained with an expectation-maximization (EM) algorithm to determine the model parameters such as mean vectors, covariance matrixes and mixture weights. And then, the log-likelihood ratio is estimated as a threshold  $\theta$  and the equal error rate (EER) is applied to evaluate the performance of GMM in test procedure. In test process of Fig.1 (b), the log-likelihood ratio  $\Lambda(X)$  estimated by the pre-trained GMMs parameters is compared with a threshold  $\theta$ . The voice is considered to be normal if  $\Lambda(X) > \theta$ , otherwise, pathological.

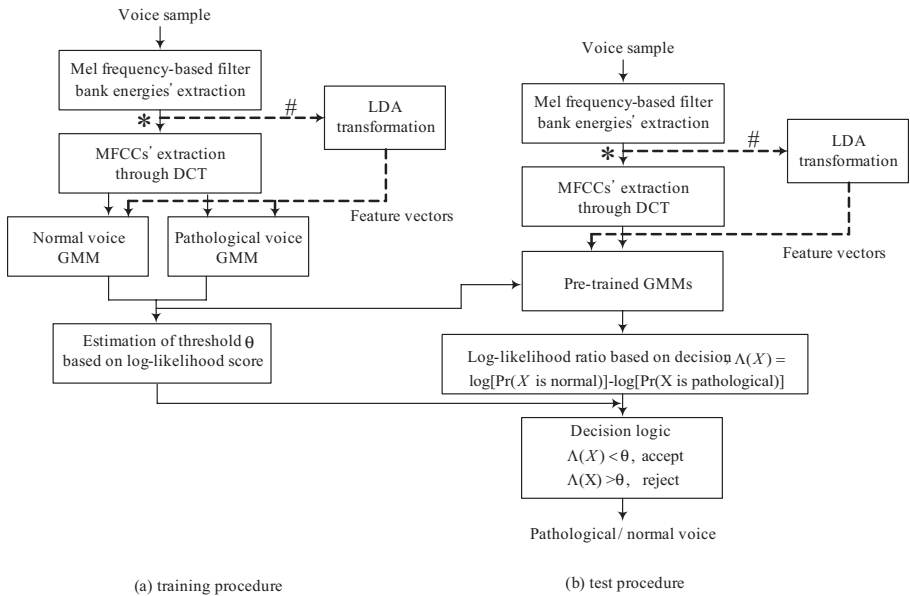


Fig. 1. Overall classification procedure

### 4.3 Effectiveness of Mel Frequency-Based Filter Bank Energies

This part demonstrates the effectiveness of the mel frequency-based filter bank energies with the analysis of the FDR[9]. The FDR has been widely used as a class separability criterion and the standard for the feature selection in speaker

recognition applications. It is defined as the ratio described in (4).

$$F_i = \frac{(\mu_{iC} - \mu_{i\bar{C}})^2}{\sigma_{iC}^2 + \sigma_{i\bar{C}}^2} \tag{4}$$

$\mu$  : class mean,  $\sigma^2$  : class variance  
 $C$  : normal voice,  $\bar{C}$  : pathological voice

This ratio selects the features which maximize a scatter between the classes. The higher the value of  $F_i$ , the more important the feature is. It means that the feature  $i$  has a low variance in regard of the inter-class variance and the feature is suitable to discriminate the classes. Fig. 2 shows the normalized FDR of the mel frequency-based filter bank energies with the 34<sup>th</sup> dimension according to the mel frequency. The usefulness of the mel frequency-based filter bank energies to classify the pathological and normal voices is found in comparatively low and high frequency bands. The largest value indicating the 1<sup>st</sup> formant appears in the low frequency band below 700 Hz. It shows that the 1<sup>st</sup> formant is the important feature to distinguish the pathological voice from normal one. Also the high frequency band above 5 kHz can be used as a discriminant feature. It tends to increase the noise at the high frequency band due to an inefficient movement of the vocal folds[12]. Those results suggest that the 1<sup>st</sup> formant and the high frequency noise are the important information to classify the pathological and normal voices. Finally, this mel frequency-based filter bank energies are converted back to the time domain MFCCs using the DCT. On the other hand, they are transformed into discriminant feature vectors through the FBE-LDA transformation. Through the FDR analysis, we can confirm that the use of the mel frequency-based filter bank energies is suitable for our purpose.

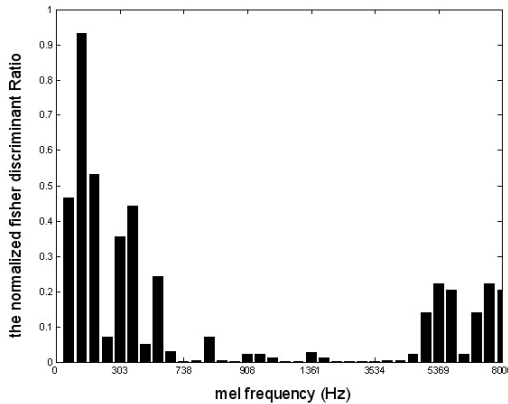


Fig. 2. The normalized FDR plot using the 34th mel frequency-based filter bank energies

#### 4.4 Baseline Performance (MFCC-Based GMM)

The GMM initialization is performed by the Linde-Buso-Gray (LBG) algorithm. Covariance matrices are diagonal. And the GMMs are trained using 2, 4, 8, 16, and 32 mixtures. In order to evaluate the performance of the voice detector, the log-likelihood ratio is compared with a threshold, the EER. The MFCCs dimension as the feature vector is 12. It is obtained from the DCT with the mel frequency-based filter bank energies ranging from the 22<sup>th</sup> to the 42<sup>th</sup> dimension. The static vectors are only used because the temporal derivatives of the MFCCs have no discriminant ability compared with the MFCCs[9]. Table 1 shows the EER of the MFCCs according to the number of the Gaussian mixtures and the number of the mel frequency-based filter bank energies. When the Gaussian mixtures are 16 and the DCT changes the 34<sup>th</sup> dimensional vector of the mel frequency-based filter bank energies into the 12<sup>th</sup> MFCCs, the best performance of the classification between pathological and normal voice, 83%, is obtained. Although the performances along with the reduction of the dimension are fairly similar, it still can be said that more number of mixtures tends to improve the performance. On the other hand, the increase in the number of the mel frequency-based filter bank energies does not tend to improve the performance.

**Table 1.** EER through MFCC-based GMM

	Mixture 2	Mixture 4	Mixture 8	Mixture 16	Mixture 32
Filter bank 22 <sup>th</sup>	25.00	22.00	20.00	18.00	18.00
Filter bank 26 <sup>th</sup>	22.00	20.00	19.00	19.00	20.00
Filter bank 30 <sup>th</sup>	21.00	21.00	20.00	18.00	20.00
Filter bank 34 <sup>th</sup>	21.00	20.00	19.00	17.00	18.00
Filter bank 38 <sup>th</sup>	21.00	21.00	21.00	20.00	21.00
Filter bank 42 <sup>th</sup>	21.00	22.00	20.00	20.00	19.00

#### 4.5 Performance of Proposed FBE LDA-Based GMM

The GMMs condition is all equal to that of the MFCC-based GMM. However, the performance improvement is measured by the several dimension reductions in the mel frequency-based filter bank energies' dimension, i.e., the 30<sup>th</sup> and the 34<sup>th</sup> show good performances in our MFCC-based experiments. Table 2 shows the EER when the mel frequency-based filter bank energies of the 30<sup>th</sup> dimension is transformed to the 12<sup>th</sup>, the 18<sup>th</sup> and the 24<sup>th</sup> dimension through the FBE-LDA transformation. As a whole, the performance is better than that of the MFCCs. The best performance is 83%. The performance has a tendency to

increase according to the number of mixtures. Table 3 shows the EER when the mel frequency-based filter bank energies of the 34<sup>th</sup> dimension is transformed to the 12<sup>th</sup>, the 18<sup>th</sup>, the 24<sup>th</sup> and the 30<sup>th</sup> dimension through the FBE-LDA transformation. As the number of mixtures and the number of features increases, the performance tends to be improved. The best performance is 85% when the mel frequency-based filter bank energies of the 34<sup>th</sup> dimension are converted into the 24<sup>th</sup> dimension and the number of mixture is 16. In conclusion, the performance is approximately improved by 2% through the FBE-LDA method. In our FBE-LDA approach, the condition for the best performance is: the features are reduced to the 24<sup>th</sup> dimension from the mel frequency-based filter bank energies of the 34<sup>th</sup> dimension and the number of mixtures is 16.

**Table 2.** EER through dimension reduction in 30<sup>th</sup> dimension’s mel frequency-based filter bank

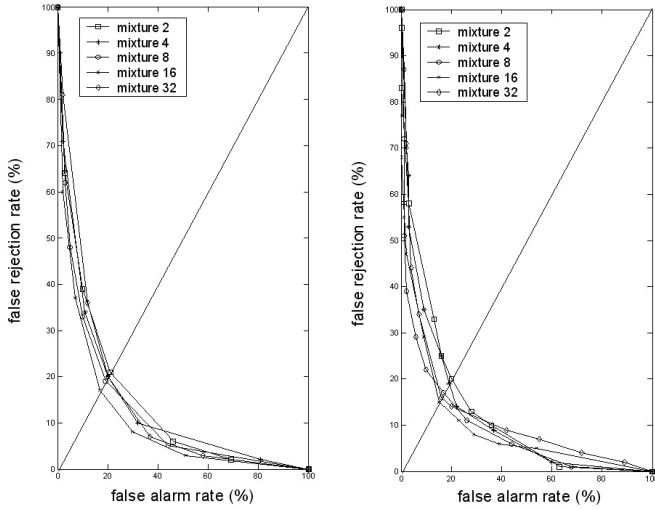
	Mixture 2	Mixture 4	Mixture 8	Mixture 16	Mixture 32
Filter bank 12 <sup>th</sup>	20.00	19.50	19.00	17.00	19.00
Filter bank 18 <sup>th</sup>	20.00	18.00	18.00	17.00	18.00
Filter bank 24 <sup>th</sup>	19.00	18.00	17.00	18.00	19.00

**Table 3.** EER through dimension reduction in 34<sup>th</sup> dimension’s mel frequency-based filter bank

	Mixture 2	Mixture 4	Mixture 8	Mixture 16	Mixture 32
Filter bank 12 <sup>th</sup>	20.00	20.00	19.00	16.00	17.00
Filter bank 18 <sup>th</sup>	20.00	19.00	18.00	17.00	17.00
Filter bank 24 <sup>th</sup>	20.00	19.00	17.00	15.00	16.00
Filter bank 30 <sup>th</sup>	20.00	19.00	18.00	17.00	17.00

#### 4.6 Comparison of Receiver Operating Characteristic (ROC) Curves

Fig.3 presents the ROC curves for the best performance in the MFCC-based GMM and FBE LDA-based GMM. Fig.3 (a) shows five different ROC curves according to the number of mixtures when the 34<sup>th</sup> dimensional vector is projected into the 12<sup>th</sup> dimensional vector space through the DCT. The best EER is 17% when the number of mixture is 16. Fig.3 (b) presents five different ROC



**Fig. 3.** ROC curves

curves according to the number of mixtures when the  $34^{th}$  dimensional vector is projected into the  $24^{th}$  dimensional vector space through the FBE-LDA transformation. The best EER is 15% when the number of mixture is 16.

## 5 Conclusion

The objective of our study is to implement FBE-LDA transformation to provide effectively discriminant feature vector classes for the pathological voice detection. And it is to compare the performances by utilizing the MFCCs and the FBE-LDA transformation with the mel frequency-based filter bank energies. We analyzed the mel frequency-based filter bank energies using the FDR and implemented the GMM detector with feature vectors through the DCT and the FBE-LDA transformation. Their relevance goes beyond the goal of a pathological voice application. Especially, there is a strong correlation between detection of pathological voice and the feature vectors through the FBE-LDA approach. The best performance is 85% when the filter bank of the  $34^{th}$  dimension is reduced to the  $24^{th}$  dimension through the FBE-LDA transformation and the number of mixtures is 16. The proposed FBE-LDA method outperforms the well-known MFCC-based GMM method. The amount of the improvement is 11.77% in an error reduction sense. This result is good compared to Godino-Llorente's research[9]. The future works may include the application and analysis of our technique in real environments and the study in the pathological type classification.

## References

1. Michaelis, D., Forhlich, M., Strobe H. W.: Selection and combination of acoustic features for the description of pathological voices. *J. Acoust. Soc. Am.* **103**(3) (1998)
2. Yingyong Qi, Robert E. Hillman, and Claudio Milstein : The estimation of signal-to-noise ratio in continuous speech for disordered voices. *J. Acoust. Soc. Am.* 105(4), April 1999
3. Vieira, M. N.: On the influence of laryngeal pathologies on acoustic and electroglottographic jitter measures. *J. Acoust. Soc. Am.* **111** (2002)
4. Hansen, J.H.L., Gavidia-Ceballos, L., and Kaiser, J.F.: A nonlinear operator-based speech feature analysis method with application to vocal fold pathology assessment. *IEEE Transactions on Biomedical Engineering*, **45**(3) (1998) 300–313
5. Childers, D.G., Sung-Bae, K.: Detection of laryngeal function using speech and electroglottographic data. *IEEE Transactions on Biomedical Engineering* **39**(1) (1992) 19–25
6. Oliveira Rosa, M., Pereira, J.C., Grellet, M.: Adaptive estimation of residual signal for voice pathology diagnosis. *IEEE Transactions on Biomedical Engineering* **47**(5) (2000) 96–104
7. Reynolds, D. A., Rose, R. C.: Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. *IEEE transaction on speech and audio processing*, **3** (1995) 72–83
8. Molla M. K. I., Hirose, K.: On the Effectiveness of MFCCs and their Statistical Distribution Properties in Speaker Identification. *IEEE international conference on VECIMS* (2004) 136–141
9. Godino-Llorente, J. I., Aguilera-Navarro, S., Gomez-Vilda, P.: Dimensionality Reduction of a Pathological Voice Quality Assessment System Based on Gaussian Mixture Models and Short-term Cepstral Parameters. *IEEE transaction on biomedical engineering*: accepted for future publication.
10. Olivier, S.: On the Robustness of Linear Discriminant Analysis as a Preprocessing Step for Noisy Speech Recognition. *Proc. IEEE conference on acoustics, speech, and signal processing* **1** (1995) 125–128
11. Kay Elemetrics Corp.: *Disordered Voice Database*. ver.1.03 (1994)
12. Kent, R. D., Ball, M. J.: *Voice Quality Measurement*. Singular Thomson Learning. (1999)

# Efficient 1D and 2D Daubechies Wavelet Transforms with Application to Signal Processing

Piotr Lipinski<sup>1</sup> and Mykhaylo Yatsymirsky<sup>2</sup>

<sup>1</sup> Division of Computer Networks, Technical University of Lodz,  
Stefanowskiego 18/22, Lodz, Poland  
piter@amuz.lodz.pl

<sup>2</sup> Department of Computer Science, Technical University of Lodz.,  
Wolczanska 215, Lodz, Poland  
jacym@ics.p.lodz.pl

**Abstract.** In this paper we have introduced new, efficient algorithms for computing one- and two-dimensional Daubechies wavelet transforms of any order, with application to signal processing. These algorithms has been constructed by transforming Daubechies wavelet filters into weighted sum of trivial filters. The theoretical computational complexity of the algorithms has been evaluated and compared to pyramidal and ladder ones. In order to prove the correctness of the theoretical estimation of computational complexity of the algorithms, sample implementations has been supplied. We have proved that the algorithms introduced here are the most robust of all class of Daubechies transforms in terms of computational complexity, especially in two dimensional case.

## 1 Introduction

Even nowadays, hardware resources necessary to process large datasets are important, and calculations often require much time, especially in two and more dimensions. Therefore, it is very important to develop more efficient computational algorithms [1].

For that reason, we still observe fast development of wavelet transforms, which resulted in many computational algorithms, to name only: recursive (Mallat's) [2], [3] modified recursive algorithm [4], lift [5], a trous [6], systolic [7], integer [8], [9], [10], and other hardware specific algorithms eg: [11], [12].

Here we focus on Daubechies [13] wavelet transforms, which have proved to be a very powerful tool in signal processing, eg: in compression [14], image analysis [15] and classification [16]. A very simple and efficient method of constructing Daubechies wavelet transform algorithms has been introduced in [17]. This method basis on transforming Daubechies wavelet filters into weighted sum of trivial filters. By applying this method to Daubechies 4 and Daubechie 6 wavelet transform, very efficient one dimensional Daubechies wavelet transform algorithms has been constructed in [17], [18]. It has been proved, that for Daubechies 4 and 6 wavelet transforms, they can double computation-saving effect when compared to the algorithms from [2-7].



In this article, we have introduced a general, efficient algorithm for calculating one- and two-dimensional Daubechies wavelet transforms of any order, based on method introduced in [17], which takes advantage of Daubechies wavelet filters transformed into weighted sum of trivial filters.

## 2 Fast Discrete Daubechies Wavelet Transforms

Here we describe the new, efficient algorithm for calculating discrete Daubechies wavelet transforms in detail. First, we analyze one dimensional case, next we extend the algorithm to two dimensions.

### 2.1 1-D Fast Discrete Daubechies Wavelet Transform

The basic idea of the Efficient Discrete Daubechies Wavelet Transform (EDDWT) algorithm is to decompose the Daubechies filters used in Mallat's decomposition scheme into a weighted sum of trivial filters. A filter is considered to be trivial if it has only absolute valued coefficients. The Daubechies Wavelet filter pair  $H_p(z)$  and  $G_p(z)$  of order  $p$ , given in [13], can be rewritten in the following, equivalent form:

$$H_p(z) = \sum_{k=0}^p \binom{p}{k} z^{-k} \cdot \sum_{k=1}^p m_{k-1} (-1)^{k-1} z^{-k+1}, \tag{1}$$

$$G_p(z) = \sum_{k=0}^p \binom{p}{k} (-1)^k z^{-k} \cdot \sum_{k=1}^p m_{p-k} z^{-k+1}, \tag{2}$$

where:

$H_n(z)$  - lowpass, orthogonal Daubechies filter,

$G_n(z)$  - highpass, orthogonal Daubechies filter,

$p$  - order of the wavelet filter,

$z$  - complex variable,

$m_p$  - is given by (3):

$$m_p = \sum_{k_1=1}^1 \sum_{k_2=2}^2 \dots \sum_{k_{p-1}=k_{p-2}+1}^{p-1} \sum_{k_p=k_{p-1}+1}^p \lambda_{k_1} \lambda_{k_2} \dots \lambda_{k_{p-2}} \lambda_{k_{p-1}}, \tag{3}$$

where:  $\lambda_k$  - is given by (4):

$$\lambda_k = \begin{cases} 1 - 2x_k + \sqrt{(1 - 2x_k)^2 - 1} & \text{when } 1 - 2x_k + \sqrt{(1 - 2x_k)^2 - 1} \leq 1 \\ 1 - 2x_k - \sqrt{(1 - 2x_k)^2 - 1} & \text{when } 1 - 2x_k + \sqrt{(1 - 2x_k)^2 - 1} > 1, \end{cases} \tag{4}$$

where  $x_k$  - is  $k$ -th root of the polynomial expressed by (5):

$$B_p(x) = \sum_{n=0}^{p-1} \binom{p+n-1}{n} x^n. \tag{5}$$

The lowpass and highpass filters given by (1) and (2) respectively are weighted sum of trivial filters, because each of the filters (6) and (7) is the product of two polynomials. The first polynomial is a trivial filter, which has only absolute coefficients  $\binom{p}{k}$ , the other is the sum of weights  $m_k$  shifted in time. The Daubechies transform can be calculated by first filtering the input signal using the trivial filters and then, adding the results multiplied by weights  $m_k$ , see fig. 1. The number of absolute multiplications can be reduced, by grouping the symmetrical coefficients and storing the outputs of trivial filters in the internal memory of a processor to avoid recalculating the same data.

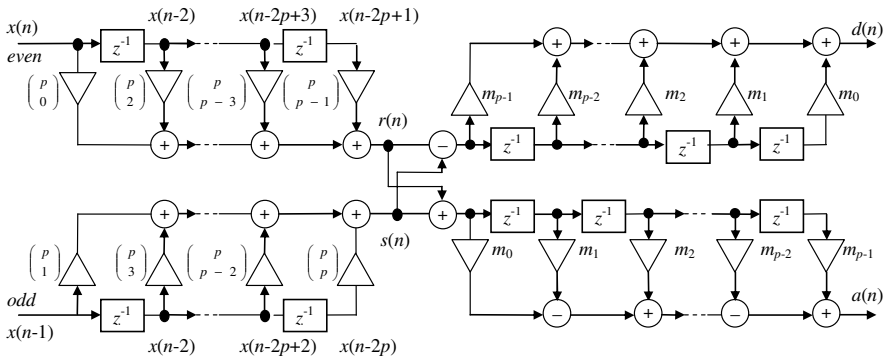


Fig. 1. Block diagram of EDDWT given by (1) and (2)

### 2.2 2-D Fast Discrete Daubechies Wavelet Transform

Two-dimensional wavelet transform is calculated recursively by applying a single step of two dimensional wavelet transform to the coarse scale approximation subband only. One step of 2D wavelet transform of order  $p$  results in four sets of data. For these four datasets, the following notation is used:  $dd_p$  (high-high or diagonal details),  $da_p$  (high-low or horizontal details),  $ad_p$  (low-high or vertical details),  $aa_p$  (low-low or approximation).  $aa_p$  subband is also called an image approximation or a coarse scale, as it represents image on a lower scale, while other subbands are referred to as image details. More computationally efficient approach involves calculating two-dimensional transform directly, by applying four 2D filters. The method of reducing the computational complexity of the wavelet transform introduced in section 2.1 can be combined with the abovementioned direct approach. The resultant algorithm will be called Efficient, Two-Dimensional, Discrete Daubechies Wavelet Transform (E2DDDWT). The four output samples of the E2DDDWT can be calculated from (6-9):

$$\begin{aligned}
 aa_p(n) = & \sum_{i=1}^p \sum_{j=i}^p m_{i-1} m_{j-1} (-1)^{i+j} \cdot (rr_p(n-i+1, n-j+1) + rs_p(n-i+1, n-j+1) + \\
 & + sr_p(n-i+1, n-j+1) + ss_p(n-i+1, n-j+1)) + \\
 & + \sum_{i=1}^p m_{m-1}^2 \cdot (rr_p(n-i+1, n-j+1) + rs_p(n-i+1, n-j+1) + \\
 & + sr_p(n-i+1, n-j+1) + ss_p(n-i+1, n-j+1)),
 \end{aligned} \tag{6}$$

$$\begin{aligned}
 ad_p(n) = & \sum_{i=1}^p \sum_{j=i}^p m_{i-1} m_{j-1} (-1)^{i-1} \cdot (rr_p(n-i+1, n-j+1) + rs_p(n-i+1, n-j+1) + \\
 & - sr_p(n-i+1, n-j+1) - ss_p(n-i+1, n-j+1)) + \\
 & \sum_{i=1}^p (-1)^{m-1} m_{m-1}^2 \cdot (rr_p(n-i+1, n-j+1) + rs_p(n-i+1, n-j+1) + \\
 & - sr_p(n-i+1, n-j+1) - ss_p(n-i+1, n-j+1)),
 \end{aligned} \tag{7}$$

$$\begin{aligned}
 da_p(n) = & \sum_{i=1}^p \sum_{j=i}^p m_{i-1} m_{j-1} (-1)^{j-1} \cdot (rr_p(n-i+1, n-j+1) - rs_p(n-i+1, n-j+1) + \\
 & + sr_p(n-i+1, n-j+1) - ss_p(n-i+1, n-j+1)) + \\
 & \sum_{i=1}^p (-1)^{m-1} m_{m-1}^2 \cdot (rr_p(n-i+1, n-j+1) - rs_p(n-i+1, n-j+1) + \\
 & + sr_p(n-i+1, n-j+1) - ss_p(n-i+1, n-j+1)),
 \end{aligned} \tag{8}$$

$$\begin{aligned}
 da_p(n) = & \sum_{i=1}^p \sum_{j=i}^p m_{i-1} m_{j-1} (-1)^{j-1} \cdot (rr_p(n-i+1, n-j+1) - rs_p(n-i+1, n-j+1) + \\
 & - sr_p(n-i+1, n-j+1) + ss_p(n-i+1, n-j+1)) + \\
 & \sum_{i=1}^p m_{m-1}^2 \cdot (rr_p(n-i+1, n-j+1) - rs_p(n-i+1, n-j+1) + \\
 & - sr_p(n-i+1, n-j+1) + ss_p(n-i+1, n-j+1)),
 \end{aligned} \tag{9}$$

where:

$m_k$  - is given by (3),

$rr_p(n,m)$ ,  $rs_p(n,m)$ ,  $sr_p(n,m)$ ,  $ss_p(n,m)$  are given by (10-13).

All the irrational multiplications required to calculate E2DDDWT appear in formulas (6-9). To compute  $rr_p(n,m)$ ,  $rs_p(n,m)$ ,  $sr_p(n,m)$ ,  $ss_p(n,m)$  we need only absolute multiplications and additions/subtractions. To reduce the number of absolute multiplications, the input values  $x(m,n)$  which are multiplied by equally-valued coefficients have been grouped together. There are three possible ways of grouping coefficients in  $rr_p(n,m)$ ,  $rs_p(n,m)$ ,  $sr_p(n,m)$ ,  $ss_p(n,m)$ : way 1 ( $p/2$  is even), way 2 ( $p/2$  is odd) and way 3 ( $p$  is odd). Due to the lack of space, only the first way is discussed in detail. Ways 2 and 3 are obtained by analogy to the way 1. by taking advantage of

extra symmetry of two-dimensional filters when  $p$  is odd or when  $p/2$  is odd.. The way 1 of grouping the coefficients is given by (10-13):

$$\begin{aligned}
 rr_p(n, m) = & \sum_{i=0}^{\lceil (p-4)/4 \rceil} \sum_{j=i}^{\lceil (p-4)/4 \rceil} \binom{p}{2i} \binom{p}{2j} \cdot (x(n-2i, m-2j) + x(n-2i-p, m-2j) + \\
 & + x(n-2i, m-2j-p) + x(n-2i-p, m-2j-p)) + \\
 & + \sum_{j=0}^{(p/4)-1} \binom{p}{p/2} \binom{p}{2j} \cdot (x(n-(p/2), m-2j) + x(n-(p/2), m-2j-p) + \\
 & + x(n-2j, m-(p/2)) + x(n-2j-p, m-(p/2))) + \left( \frac{p}{p/2} \right)^2 x(p/2, p/2),
 \end{aligned} \tag{10}$$

$$\begin{aligned}
 rs_p(n, m) = & \sum_{i=0}^{\lceil (p-4)/4 \rceil} \sum_{j=i}^{\lceil (p-6)/4 \rceil} \binom{p}{2i+1} \binom{p}{2j} \cdot (x(n-2i, m-2j-1) + x(n-2i-p, m-2j-1) + \\
 & + x(n-2i, m-2j-p+1) + x(n-2i-p, m-2j-p+1)) \\
 & + \sum_{i=0}^{(p/4)-1} \binom{p}{p/2} \binom{p}{2i+1} \cdot (x(n-(p/2), m-2i-1) + x(n-(p/2), m+2i-p+1)),
 \end{aligned} \tag{11}$$

$$\begin{aligned}
 sr_p(n, m) = & \sum_{i=0}^{\lceil (p-6)/4 \rceil} \sum_{j=i}^{\lceil (p-6)/4 \rceil} \binom{p}{2j+1} \binom{p}{2i} \cdot (x(n-2i-1, m-2j) + \\
 & x(n-2i-p+1, m-2j) + x(n-2i-1, m-2j-p) + x(n-2i-p+1, m-2j-p)) + \\
 & + \sum_{j=0}^{p/2} \binom{p}{p/2} \binom{p}{2j+1} \cdot (x(m-2j-1, n-(p/2)) + x(m+2j+1-p, n-(p/2))),
 \end{aligned} \tag{12}$$

$$\begin{aligned}
 ss_p(n, m) = & \sum_{i=0}^{\lceil (p-6)/4 \rceil} \sum_{j=i}^{\lceil (p-6)/4 \rceil} \binom{p}{2j+1} \binom{p}{2i} \cdot (x(n-2i-1, m-2j-1) + \\
 & + x(n-2i-p+1, m-2j-1) + x(n-2i-1, m-2j-p+1) + \\
 & + x(n-2i-p+1, m-2j-p+1)).
 \end{aligned} \tag{13}$$

The E2DDDWT is listed as follows: First, find the value of the parameter  $p$ . Second, chose appropriate formulas to calculate  $rr_p(n, m)$ ,  $rs_p(n, m)$ ,  $sr_p(n, m)$ ,  $ss_p(n, m)$ : (10-13) or those, corresponding to case 2 or 3. Last, calculate  $aa_p(n, m)$ ,  $ad_p(n, m)$ ,  $da_p(n, m)$ ,  $dd_p(n, m)$  from  $rr_p(n, m)$ ,  $rs_p(n, m)$ ,  $sr_p(n, m)$ ,  $ss_p(n, m)$ .

### 3 Computational Complexity

Here, the computational complexity of the algorithms introduced in section 2 is discussed and compared to the Mallat's and Lift algorithms. In order to make the comparison exhaustive, the number of irrational multiplications, rational multiplications and additions/subtractions, required to compute a single step of the Daubechies wavelet transform of any order is compared. First, we discuss the number of arithmetic operations required to compute a single step of 1D Daubechies wavelet

transform. Here we assume that a single step of 1D Daubechies wavelet transform is equivalent to calculating two output samples (lowpass and highpass) out of two input samples.

The total number of irrational multiplications, additions/subtractions, absolute multiplications required to calculate one step of E2DDDWT, using the following three algorithms: Mallat's, lift, and EDDWT, are compared in table 1.

**Table 1.** The number of irrational multiplications (imul), absolute multiplications (amul) and additions/subtractions (add) required to calculate two output samples of a single step of a DWT of order  $p$

p		Mallat			Lift			EDDWT		
		imul	amul	add	imul	amul	add	imul	amul	add
db 4	2	8	-	6	5	-	4	2	2	12
db 6	3	12	-	10	8	-	6	4	4	12
db 2p	p even	4p	-	4p-2	$\approx 2p^*$	-	$\approx 2p-1$	$2(p-1)(p-1)/2$		3p-1
db 2p	p odd	4p	-	4p-2	$\approx 2p^*$	-	$\approx 2p-1$	$2(p-1)$	p/4	3p-1

\* for  $p \rightarrow \infty$

In 2-D algorithm, we assume that a single step is equivalent to calculating four output samples out of four input samples. The number of irrational multiplications, additions/subtractions and absolute multiplications required to compute a single step of two dimensional Daubechies wavelet transforms using: Mallat's, lift and E2DDDWT are compared in table 2.

**Table 2.** The number of irrational multiplications (imul), absolute multiplications (amul) and additions/subtractions (add) required to calculate four output samples of a single step of E2DDDWT of order  $p$

p		Mallat			Lift			EDDWT		
		imul	amul	add	imul	amul	add	imul	amul	add
db 4	2	64	-	60	100	-	64	8	-	12
db 6	3	144	-	100	256	-	144	20	-	64
db 2p	p even	$16p^2$	-	$4(2p-1)^2$	$\approx 16p^2$	-	$4(2p-1)^2$	$2(p^2+p)-4p^2/2$	$+3p+4$	$6p^2+2p+16$
db 2p	p odd	$16p^2$	-	$4(2p-1)^2$	$\approx 16p^2$	-	$4(2p-1)^2$	$2(p^2+p)-4$	$4(p+1)^2$	$6p^2+2p+16$

The number of absolute multiplications depends on the parity of the coefficient  $p$ . If  $p$  is even, the number of absolute multiplications is equal to  $p^2/2+3p+4$ , otherwise (non symmetrical coefficients) the number of multiplications is equal to  $4(p+1)^2$ . This is due to the symmetry of two dimensional filters, mentioned in section 2.

Notice, that calculating four output samples:  $aa_p(n,m)$ ,  $ad_p(n,m)$ ,  $da_p(n,m)$ ,  $dd_p(n,m)$ , requires only the values of corresponding:  $rr_p(i,j)$ ,  $rs_p(i,j)$ ,  $sr_p(i,j)$ ,  $ss_p(i,j)$ , only for  $(n,m)$   $(n,m-1)$   $(n-1,m)$   $(n-1,m-1)$ , because the values of:  $rr_p(i,j)$ ,  $rs_p(i,j)$ ,  $sr_p(i,j)$ ,  $ss_p(i,j)$ , for  $i < n-1$  and  $j < m-1$  have already been calculated in previous steps. For example,  $rr_p(n,m-2)$ , has already been calculated in for  $aa_p(n,m-2)$ .

Low-level computational complexity of irrational multiplications, absolute multiplications and additions/subtractions are different. Furthermore, they strongly depend on hardware and software implementations. Here we assume that irrational multiplication is 10 times the computational complexity of addition/subtraction and the absolute multiplication is twice the computational complexity of addition/subtraction. The number of low level operations required to calculate EDDWT and E2DDDWT are given in table 3.

**Table 3.** The number of low level operations required to calculate two output samples of a single step of EDDWT and four output samples of a single step of E2DDDWT of order  $p$

		1-D			2-D		
	p	Mallat	Lift	EDDWT	Mallat	Lift	E2DDDWT
db 4	2	86	54	34	700	1064	148
db 6	3	130	86	56	1540	2704	404
db 2p	p even	$44p-2$	$\approx 22p-1$	$24p-22$	$176p^2-16p+4$	$\approx 176p^2-16p+1$	$27p^2+28p-16$
db 2p	p odd	$44p-2$	$\approx 22p-1$	$23,5p-22$	$176p^2-16p+4$	$\approx 176p^2-16p+1$	$34p^2+38p-16$

To validate the estimations given in table 3, we have implemented the Mallat and EDDWT algorithm in Assembler 68000, compiled using asm68k and executed on MC68EC030 microprocessor. The number of clock cycles required to calculate a wavelet transform of 12,8kB of 16-bit fixed-point samples has been compared. A single step of Daubechies 4 transform calculated using Mallat's transform required about 900 thousands clock cycles, while using EDDWT required only about 300 thousands clock cycles. A single step of Daubechies 6 Mallat's algorithm required about 1400 thousands clock cycles, while EDDWT only about 600 thousands. The abovementioned results have proved the correctness of the estimations given in table 3.

## 4 Conclusion

In this paper we have introduced new efficient, algorithms for one- and two-dimensional Daubechies wavelet transform computation with application signal processing. We have provided the detailed algorithm description together with theoretical comparison of the computational complexity of the efficient algorithms with the most popular ones: Mallat's and Lift. The comparison have proved that, the one dimensional Daubechies wavelet transform (EDDWT) algorithm introduced here has lower computational complexity (up to 50%) than Mallat's and slightly lower computational complexity than Lift algorithms. In two dimensional case (images) the E2DDDWT outperforms well known algorithms in terms of computational complexity, as it can be even four times faster than Mallat and Lift. Implementation on MC68EC030 microprocessor proved the correctness of theoretical estimations of computational complexity and in turn proved the efficiency of one- and two-dimensional Daubechies wavelet algorithms introduced here.

## References

1. Ayman A., et al.: Mammogram Image Size Reduction Using 16-8 bit Conversion Technique. *International Journal Of Biomedical Sciences* Vol. 1 No. 2 (2006)
2. Mallat, S.: *A Wavelet Tour of Signal Processing*. Academic Press (1988)
3. Vishwanath M.: The recursive pyramid algorithm for the discrete wavelet transform, *IEEE Trans. Signal Process.*, vol. 42, no. 3, Mar. (1994), 673–677
4. Chakrabarti Ch., Vishwanath M.: Efficient Realizations of the Discrete and Continuous Wavelet Transforms: From Single Chip Implementations to Mappings on SIMD Array Computers, *IEEE Transactions On Signal Processing*, Vol. 43, No. 3, March (1995)
5. Sweldens W.: The lifting scheme: A construction of second generation wavelets, *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, (1998) 511–546
6. Holschneider, M., Kronland-Martinet R, Morlet J., *Wavelets, Time-Frequency Methods and Phase Space*, chapter A: Real-Time Algorithms for Signal Analysis with the Help of the Wavelet Transform. Springer Verlag. Berlin (1989) 289-297
7. Vishwanath M., Owens R. M., Irwin M. J.: VLSI Architectures for the Discrete Wavelet Transform, *IEEE Transactions On Circuits And Systems-11: Analog And Digital Signal Processing*, Vol. 42, No. 5. May (1995)
8. Mao J. S., Chan S. C., Liu W., Ho K. L., Design and multiplierless implementation of a class of two-channel PR FIR filterbanks and wavelets with low system delay, *IEEE Trans. Signal Processing*, vol. 48, Dec. (2000) 3379–3394
9. Akansu A. N.: Multiplierless PR quadrature mirror filters for subband image coding, *IEEE Trans. Image Processing*, vol. 5, Sept. (1996) 1359–1363
10. Kotteri K. A., Bell A.E., Carletta J.E.: Design of Multiplierless, High-Performance, Wavelet Filter Banks With Image Compression Applications *IEEE Transactions On Circuits And Systems I: Regular Papers*, vol. 51, No. 3, March (2004)
11. Bayoumi M. A.: Three-Dimensional Discrete Wavelet Transform Architectures Michael Weeks, *IEEE Transactions On Signal Processing*, Vol. 50, No. 8, August (2002)
12. Barua S., Carletta J.E., Kotterib K.A.: An efficient architecture for lifting-based two-dimensional discrete wavelet transforms, *Integration, The Vlsi Journal* 38, (2005) 341–352
13. Daubechies I, *Ten Lectures on Wavelets*, SIAM, (1992) 357-367
14. Kuduvalli G. R., Rangayyan R. M.,: Performance analysis of reversible image compression techniques for high resolution digital teleradiology," *IEEE Trans. Med. Imag.*, vol. 11, (1992) 430-445
15. Lina J.M.: Image Processing with Complex Daubechies Wavelets, *Journal of Mathematical Imaging and Vision* 7, (1997) 211–223
16. Patnaik L.M.: Daubechies 4 wavelet with a support vector machine as an efficient method for classification of brain images. *Journal of Electronic Imaging* Vol. 14, Issue 1,-January - March (2005)
17. Lipiński P.: Fast Algorithm For Daubechies Discrete Wavelet Transform Computation. *Nacjonalna Akademia Nauk Ukrainy, Modelowanie i Technologie Informacyjne, Zbiór prac naukowych* No. 19, Kiev (2002) 178–183
18. Lipiński P.: Optimized 1-D Daubechies 6 Wavelet Transform / *Information Technologies and Systems*, Vol. 6, No 1-2, (2003) 94-98

# A Branch and Bound Algorithm for Matching Protein Structures

Janez Konc and Dušana Janežič

National Institute of Chemistry, Hajdrihova 19, SI-1000 Ljubljana, Slovenia  
{konc, dusa}@cmm.ki.si

**Abstract.** An efficient branch and bound algorithm for matching protein structures has been developed. The compared protein structures are represented as graphs and a product graph of these graphs is calculated. The resulting product graph is then the input to our algorithm. A maximum clique in the product graph corresponds to the maximum common substructure in the original graphs. Our algorithm, which gives an approximate solution to the maximum clique problem, is compared with exact algorithms commonly used in bioinformatics for protein structural comparisons. The computational results indicate that the new algorithm permits an efficient protein similarity calculation used for protein structure analysis and protein classification.

## 1 Introduction

A clique is a subset of vertices in a graph  $G$  such that each pair of vertices in this subset is connected by an edge. The maximum clique problem is the problem of finding in a given graph the clique with the largest number of vertices. A maximal clique is a clique that is not a subset of a larger clique. The algorithms for finding cliques are frequently used in bioinformatics, where these algorithms have been applied to compare three-dimensional molecular structures [1]. Among these algorithms is the widely used Bron and Kerbosch algorithm [2]. However, several other algorithms have also been used [3]. Searching for the maximum clique is often the bottle-neck computational step in these applications, as the maximum clique problem is NP-hard [4].

Exact algorithms, which can be guaranteed to find the maximum clique, usually use a branch-and-bound approach to the maximum clique problem [3,5], searching systematically through the possible solutions and applying bounds to limit the search space. The tightest bounds come from the vertex-coloring method. This method assigns colors to vertices in a way that no two adjacent vertices of a graph  $G$  are colored with the same color. The number of colors is the upper bound to the size of the maximum clique in graph  $G$ . Vertex-coloring is also known to be NP-hard [4], therefore a graph can only be colored approximately. Considering the nature of the problem can provide more efficient algorithms.

In this paper we present a new branch and bound algorithm *MatchProt*, which combines a maximum clique algorithm with an approach inspired by the geometric



hashing method. Geometric hashing is a widely used method for structural comparisons and pattern recognition [6]. While searching for a maximum clique, our algorithm takes advantage of the three-dimensionality of the compared objects. We apply our algorithm to the problem of searching for similarities in the structures of biological macromolecules, a problem that often presents itself in protein classification. We compare our algorithm with the algorithm of Tomita and Seki and with the algorithm of Bron and Kerbosch. We show that our algorithm is slightly faster than the maximum clique algorithm of Tomita and Seki, but is significantly faster than the maximal clique algorithm of Bron and Kerbosch. In most cases our algorithm finds larger similarities than the above listed exact algorithms.

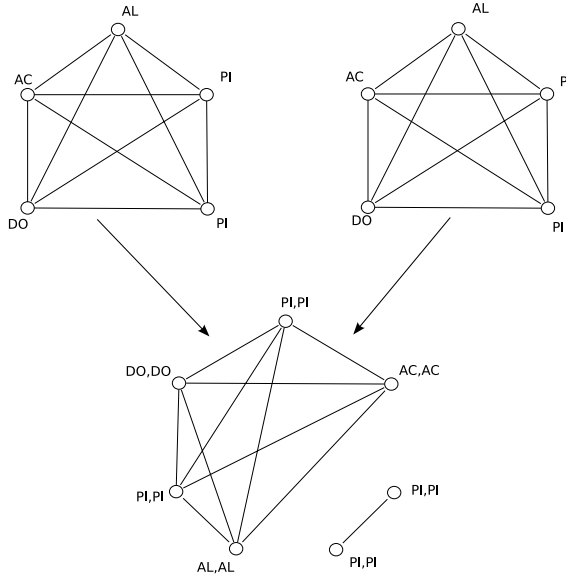
## 2 Theory

**Notations.** An undirected graph  $G = (V, E)$  consists of a set of vertices  $V = \{1, 2, \dots, n\}$  and a set of edges  $E \subseteq V \times V$ . Two vertices  $v$  and  $w$  are adjacent, if there exists an edge  $(v, w) \in E$ . For a vertex  $v \in V$ , a set  $\Gamma(v)$  is the set of all vertices  $w \in V$  that are adjacent to the vertex  $v$ .  $|\Gamma(v)|$  is the degree of vertex  $v$ . The maximum degree in  $G$  is denoted as  $\Delta(G)$ . Let  $G(R) = (R, E \cap R \times R)$  be the subgraph induced by vertices in  $R$ , where  $R$  is a subset of  $V$ . The density of a graph is calculated as  $D = |E|/(|V| \cdot (|V| - 1)/2)$ . The number of vertices in a maximum clique is denoted by  $\omega(G)$ .

**Protein graph.** Three-dimensional objects that are to be compared with the clique algorithms, must be represented as graphs of vertices and edges. Vertices are points with labels in three-dimensional space. An edge is inserted between any two vertices, if they conform to a chosen criterion. In this work we encode proteins as graphs by a set of rules adopted from Scmitt et al. [1]. Five labels, hydrogen bond donors (DO), hydrogen bond acceptors (AC), mixed acceptors/donors (ACDO), aromatic (PI), and aliphatic (AL), describe interactions of functional groups that may occur on the protein surface. Functional groups of surface residues are represented by points in space, with their corresponding labels, which together form vertices of a protein graph. If two vertices are less than *cutoff* Angstroms apart, then an edge is drawn between these two vertices.

**Product graph.** Two protein graphs are used to construct a product graph. A maximum clique in this graph corresponds to the maximum substructure that is common to both graphs [7]. A vertex in a product graph is a pair of vertices  $(m, n)$ , where the first member, vertex  $m$ , belongs to the first protein graph, and the second member, vertex  $n$ , belongs to the second protein graph and the labels of the two vertices  $m$  and  $n$  match. An edge between two vertices  $(m, n)$  and  $(p, q)$  is drawn if the difference between distances  $d(m, p) - d(n, q) < resolution$ . In our study we use typical values for the two parameters, *cutoff* being 12.0 – 30.0 and *resolution* being 2.0Å. The examples of two protein graphs, together with their product graph, are shown in Fig. 1.

**The MatchProt algorithm.** A maximum clique in a product graph constructed with the above listed rules is equivalent to a rigid body rotation and translation of



**Fig. 1.** A product graph is constructed from two protein graphs and the maximum clique in this product graph,  $Q_{max} = \{(AC, AC), (DO, DO), (PI, PI), (PI, PI), (AL, AL)\}$  corresponds to the maximum common substructure in the two original graphs

the vertices of the first to the vertices of the second graph that aligns the most vertices. In three dimensions, any three product graph vertices which are in a clique and correspond to the best alignment of three vertices of the first to three vertices of the second protein graph, are enough to define this geometrical transformation. Because we search for the transformation that would align the most vertices in three-dimensional proteins, we search for cliques with three vertices, using an efficient branch and bound algorithm for finding a maximum clique in an undirected graph [3,8]. When we find such a clique, we first calculate the rotation and translation of the first protein graph to the second protein graph, based on the transformation defined by this three clique vertices, and second, we append to this clique other vertices of the two protein graphs that this transformation has aligned. The pseudocode for the *MatchProt* algorithm is shown in Fig. 2

The algorithm *MatchProt* maintains two global sets,  $Q$  and  $Q_{max}$ , where  $Q$  consists of vertices of the currently growing clique and  $Q_{max}$  consists of vertices of the largest clique currently found. The algorithm starts with an empty set  $Q$ , and then recursively adds vertices from the product graph to (and deletes vertices from) this set, until it can verify that no clique with more vertices can be found. The next vertex to be added to  $Q$  is selected from the set of candidate vertices  $R \subseteq V$ , which is initially set to  $R := V$ , where  $V$  is the set of vertices of the product graph. At each step, the algorithm selects a vertex  $p \in R$  with the maximum color  $C(p)$  among the vertices in  $R$ , and deletes it from  $R$ .  $C(p)$  is the upper bound to the size of the maximum clique in the resulting set  $R$ . If

```

Procedure MatchProt( $R, C$ )
  while  $R \neq \emptyset$  do
    choose a vertex  $p$  with a maximum color  $C(p)$  from set  $R$ ;
     $R := R \setminus \{p\}$ ;
    if  $|Q| + C(p) > |Q_{max}|$  then
       $Q := Q \cup \{p\}$ ;
       $R_p := R \cap \Gamma(p)$ ;
      if  $|Q| \geq 3$  then
        find best superposition of vertex pairs in  $Q$ ;
        rotate and translate pairs of vertices in  $R_p$  accordingly;
         $P :=$  vertices in  $R_p$  that are best aligned;
         $R_p := R_p \cap \Gamma(P)$ ;
         $Q := Q \cup P$ ;
      if  $R_p \neq \emptyset$  then
        obtain a vertex-coloring  $C'$  of  $G(R_p)$ ;
        MatchProt( $R_p, C'$ );
      else if  $|Q| > |Q_{max}|$  then  $Q_{max} := Q$ ;
       $Q := Q \setminus P \setminus \{p\}$ ;
    else return
  end while

```

**Fig. 2.** The *MatchProt* algorithm

the sum  $|Q| + C(p)$  indicates that a clique larger than the one currently in  $Q_{max}$  can be found in  $R$ , then vertex  $p$  is added to the set  $Q$ . The new candidate set  $R_p = R \cap \Gamma(p)$  is calculated.

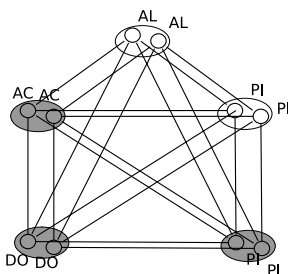
If  $|Q| \geq 3$ , then the algorithm finds the best superposition of the first protein graph to the second protein graph according to  $Q$ , so that all first member vertices  $m$  in product graph vertices  $(m, n) \in Q$  are superposed to their corresponding second member vertices  $n$ . For all pairs of product graph vertices in  $R_p$ , we calculate the distance  $d(v, w)$  between the coordinates of the vertices in a pair  $(v, w)$ . If this distance is less than the *resolution* = 2.0Å, and is minimal between all distances in which a vertex  $v$  or a vertex  $w$  appear, then we remove this pair of vertices from  $R_p$  and add it to the set  $P$ . All neighbors of vertices in the set  $P$  are also removed from  $R_p$ . All vertices from the set  $P$  are then added to the clique  $Q$ .

A vertex-coloring  $C'$  is then calculated [3,8] and passed as a parameter to the recursive call of the *MatchProt* procedure. If  $R_p = \emptyset$  and  $|Q| > |Q_{max}|$ , *i.e.*, the current clique is larger than the currently largest clique found, then the vertices of  $Q$  are copied to  $Q_{max}$ . The algorithm then backtracks by removing  $p$  from  $Q$  and then selects the next vertex from  $R$ . This procedure continues until  $R = \emptyset$ . Table 1 shows recursive steps of the *MatchProt* algorithm, taking as the input the vertices and the edges of the product graph from Fig. 1. When three clique vertices are found (step 5\* in Table 1, grey ovals in Fig. 3), the two protein graphs are superposed. The two aligned vertices of the two protein graphs (step 5\* in Table 1, transparent ovals in Fig. 3) are appended to the clique in a single step (step 5), and the new maximum clique is stored in  $Q_{max}$ . For the superposition of vertices we use an algorithm from the literature [9].

**Table 1.** Recursive steps of the *MatchProt* algorithm taking as the input the product graph from Fig. 1.  $Q$  is the set for storing clique vertices,  $P$  is the set of vertices that are aligned by the superposition, and  $Q_{max}$  is the set of vertices of the currently largest clique found

Step	$Q$	$P$	$Q_{max}$
1	PI <sub>2</sub>	∅	∅
2	PI <sub>2</sub> , PI <sub>2</sub>	∅	PI <sub>2</sub> , PI <sub>2</sub>
3	AC <sub>2</sub>	∅	–
4	AC <sub>2</sub> , DO <sub>2</sub>	∅	–
5*	AC <sub>2</sub> , DO <sub>2</sub> , PI <sub>2</sub>	AL <sub>2</sub> , PI <sub>2</sub>	–
5	AC <sub>2</sub> , DO <sub>2</sub> , PI <sub>2</sub> , AL <sub>2</sub> , PI <sub>2</sub>	∅	AC <sub>2</sub> , DO <sub>2</sub> , PI <sub>2</sub> , AL <sub>2</sub> , PI <sub>2</sub>
⋮	⋮	⋮	⋮

AC<sub>2</sub>, DO<sub>2</sub>, PI<sub>2</sub> and AL<sub>2</sub> is used instead of (AC,AC), (DO,DO), (PI,PI) and (AL,AL).  
 \*indicates the superposition step depicted in Fig. 3



**Fig. 3.** The superposition of the two example protein graphs from Fig. 1. Encircled and in grey are the vertices of the two protein graphs, which are already a part of the clique, and are used for calculating the superposition of the two graphs. Encircled are the two aligned vertices, which will be added to the clique.

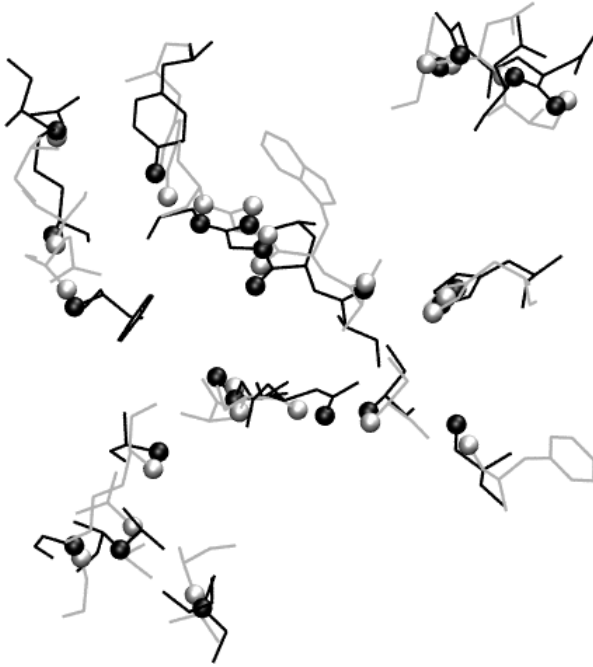
**Initialization.** We set  $Q := \emptyset$ ,  $Q_{max} := \emptyset$ . We calculate the degrees of vertices  $V$  in graph  $G$  and sort these vertices in a non-increasing order with respect to these degrees. The first  $\Delta(G)$  vertices in  $V$  are colored with numbers  $1 \dots \Delta(G)$  and the rest of the vertices in  $V$  are assigned a color  $\Delta(G)$ . The first candidate set  $V$ , together with its coloring  $C$ , is then an input to the *MatchProt* procedure. This initialization follows the standard procedure described elsewhere [3].

### 3 Results

To evaluate the ability of our algorithm to detect similarities in protein structures we have taken two proteins with PDB codes 1TPO and 2PRK. These two proteins share a similar binding site, although their sequence similarity is low. We have extracted the surface residues around the binding site of each of the two proteins [10] and encoded them as graphs according to the above listed

**Table 2.** The known similarity in proteins with PDB codes 1TPO and 2PRK

1TPO		2PRK	
Residue	#	Residue	#
HIS	57	HIS	69
HIS	57	HIS	69
SER	214	SER	132
SER	217	GLY	135
GLY	216	GLY	134
GLY	216	GLY	134
TRP	215	LEU	133
CYS	191	ASN	161
GLN	192	THR	223
CYS	191	GLY	160

**Fig. 4.** The superposition of the binding sites of the proteins with PDB codes 1TPO and 2PRK. Protein 1TPO is colored black and protein 2PRK is grey. Black and grey points correspond to aligned vertices of the two protein graphs.

rules. Then we constructed a product graph from the two labelled graphs, which was the input to our algorithm. The aligned binding site residues are shown in Table 2 and the obtained superposition of these residues is shown in Fig. 4. The *MatchProt* algorithm (MP) was compared with the Tomita and Seki algorithm (TS) and with the well-known Bron and Kerbosch algorithm (BK), the latter

**Table 3.** CPU times [s] and numbers of steps for input product graph of size  $N$  around the binding site center of proteins with PDB codes 1TPO and 2PRK; *cutoff* in Angstroms. The *MatchProt* algorithm (MP) is compared to Tomita and Seki algorithm (TS) and to the Bron and Kerbosch (BK) algorithm.  $\omega$  is the size of the maximum common subgraph.

Graph		MP			TS			BK		
<i>cutoff</i>	N	$\omega$	#steps	CPU time	$\omega$	#steps	CPU time	$\omega$	#steps	CPU time
12.0	296	10	305	0.0044	10	317	0.0045	10	8865	0.0078
15.0	655	13	792	0.023	13	795	0.022	13	82147	0.071
18.0	957	17	1197	0.050	16	1405	0.055	16	372846	0.33
21.0	1182	19	1363	0.082	18	1728	0.092	18	983539	0.87
24.0	1470	19	2711	0.16	20	2913	0.18	20	2743414	2.44
27.0	1851	24	3508	0.31	22	7032	0.42	22	7379746	6.52
30.0	2435	24	11912	0.92	23	16330	1.13	23	24510454	21.95

being the algorithm of choice for many applications where biomolecular structures are compared. All calculations were performed on an 1.6GHz Opteron processor.

The results of the comparisons are shown in Table 3. We find that the MP algorithm needs less steps than the TS algorithm to find an approximately maximum common substructure in the two test proteins. The calculation time is also modestly reduced in comparison with the MP algorithm, however we believe that some improvements to our algorithm are still possible. The MP algorithm generally finds larger common substructures ( $\omega$ ) than the two exact algorithms. The MP and TS algorithms are both significantly faster and need less steps than the BK algorithm, which is due to the fact that the latter algorithm searches for all maximal cliques in a graph instead of only searching for the maximum one.

## 4 Conclusions

In this paper we describe an efficient branch and bound algorithm for comparing protein structures. Our algorithm uses bounds provided by a maximum clique finding algorithm and combines it with a geometric approach to take advantage of the specific three-dimensional structure of protein graphs. With this approximation it is possible to reduce the time to find the maximum common substructure. Our algorithm is considerably faster than a widely used algorithm. Its use is likely in protein structural comparisons [11] and protein classifications.

**Acknowledgements.** The financial support through grant No. P1-0002 of the Ministry of Higher Education, Science, and Technology of Slovenia is acknowledged.

## References

1. Schmitt, S., Kuhn, D., Klebe, G.: A new method to detect related function among proteins independent of sequence and fold homology. *Journal of Molecular Biology* **323** (2002) 387–406
2. Bron, C., Kerbosch, J.: Algorithm 457 - Finding all cliques of an undirected graph. *Commun. ACM* **16** (1973) 575–577
3. Tomita, E., Seki, T.: An efficient branch-and-bound algorithm for finding a maximum clique. *Lecture Notes in Computer Science* **2631** (2003) 278–289
4. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
5. Östergård, P.R.J.: A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics* **120** (2002) 197–207
6. Pennec, X., Ayache, N.: A geometric algorithm to find small but highly similar 3D substructures in proteins. *Bioinformatics* **14** (1998) 516–522
7. Raymond, J.W., Willett, P.: Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design* **16** (2002) 521–533
8. Konc, J., Janežič, D.: A maximum clique problem revisited. *European Journal of Operations Research* (to appear)
9. Kabsch, W.: A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Cryst.* **A34** (1978) 827–828
10. Konc, J., Hodošček, M., Janežič, D.: Molecular surface walk. *Croat. Chem. Acta* **79** (2006) 237–241
11. Kristan, K., Krajnc, K., Konc, J., Stanislav, G., Stojan, J.: Phytoestrogens as inhibitors of fungal  $17\beta$ -hydroxysteroid dehydrogenase. *Steroids* **70** (2005) 626–635

# Multimodal Hand-Palm Biometrics

Ryszard S. Choraś and Michał Choraś

Image Processing Group, Institute of Telecommunications,  
University of Technology & Life Sciences,  
S. Kaliskiego 7, 85-791 Bydgoszcz, Poland  
{choras, chorasm}@utp.edu.pl

**Abstract.** Hand geometry based biometric verification has proven to be the most suitable and acceptable biometrics trait for medium and low security applications. Hereby a new approach for the personal identification using hand images is presented. Two kinds of biometric indicators are extracted from the low-resolution hand images; (i) palmprint features, which are composed of principal lines, wrinkles, minutiae, delta points, etc., and (ii) hand geometry features which include area/size of palm, length and width of fingers. In the article we focus on feature extraction methods applied to one-sensor multimodal hand-palm biometrics system.

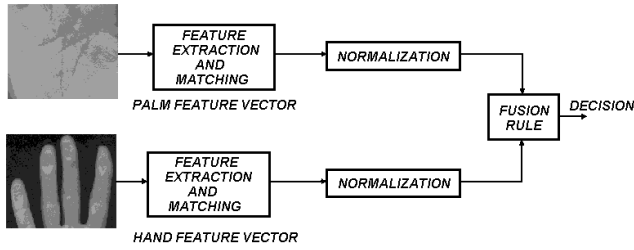
## 1 Introduction

Multimodal biometrics has recently gained much attention and popularity in the computer science society mainly because it enables to improve identification rates and can overcome limitations of single biometrics systems. Hands and palmprints are popular biometrics characteristics and it seems natural to merge them together in one-sensor multimodal identification system.

Therefore in the article we propose a robust, accurate method for personal identification based on palmprint features and hand shape geometry. The method consists of palm feature extraction, hand feature extraction and feature classification steps (Figure 1). In the hand feature extraction step, we define the basic hand geometry as the contour of the hand, finger tips, and finger "valleys". Generally, contour extraction is a similar problem as contour extraction in computer vision. In our case a simple Canny edge algorithm is enough to extract a proper contour line. Next, landmark points are extracted by searching curvature extremities and corner point detection and we calculate eight area features describing hand geometry. The proposed palmprint feature detection method uses Zernike Moment Invariants [1], [2].

The article is organized as follows: in Section 2 we overview existing related literature, in Section 3 we present our feature extraction methods from hand and palm images. In Section 4 fusion rules, performed experiments and achieved results are discussed, while the conclusion is given next.





**Fig. 1.** Block diagram of the biometric system using palm and hand geometry

## 2 Related Work

Hand-based authentication schemes in the literature are mostly based on geometrical features. For example, Sanchez-Reillo et al. [3] measured finger widths at different latitudes, finger and palm heights, finger deviations and the angles of the inter-finger valleys with the horizontal. The twenty-five selected features were modelled with Gaussian mixture models specific to each individual. Jain, Ross and Pankanti [4] used a peg-based imaging scheme and obtained sixteen features, which include length and width of the fingers, aspect ratio of the palm to fingers, and thickness of the hand. The prototype system they had developed was tested in a verification experiment for web access over for a group of 10 people. Bulatov et al. [5] extracted geometric features similar to [3], [4], [6] and compared two classifiers. Lay [7] introduced a technique where the hand is illuminated with a parallel grating that serves both to segment the background and enables the user to register his hand with one the stored contours. The geometric features of the hand shape were captured by the quadtree code. Saeed and Werdoni proposed to use minimal eigenvalues of Toeplitz matrices for hand recognition [8].

Palmprint feature extraction methods are mainly based on geometrical parameters, lines topology, texture features, Wavelets and Fourier transforms etc. [9], [10], [11], [12], [13], [14], [15].

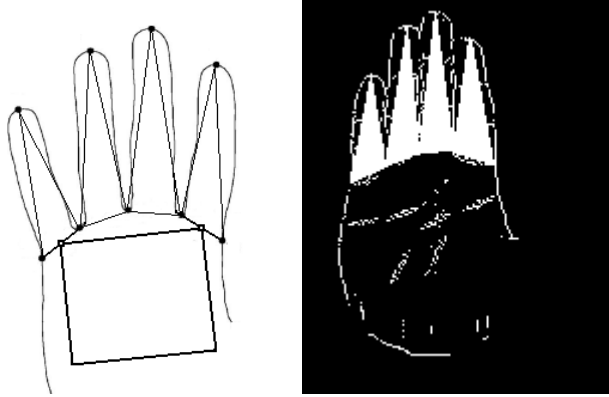
## 3 Feature Extraction Methods in Multimodal Hand-Palm Biometrics

### 3.1 Hand Geometry Feature Extraction

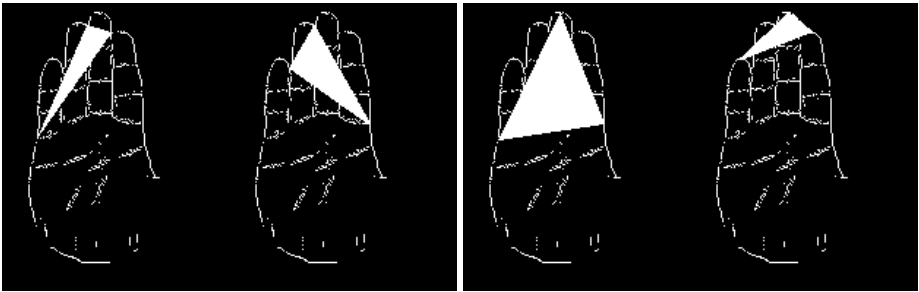
Hand geometry refers to the geometric structure of hand, which includes lengths of fingers, widths at various points on the finger, diameter of the palm, thickness of the palm, etc. These features are not as discriminating as other biometric characteristics (such as fingerprints), however they can easily be used for verification purpose.

To obtain these features, an image of the silhouetted hand is needed. In this paper, we regard a roof edge as two step edges and use a step edge detection

algorithm such as Canny's algorithm to detect it [16]. Since there is clear distinction in intensity between the hand and the background by design, a binary image is obtained through thresholding and hand boundary is easily located afterwards. Geometrical landmarks, i.e. the fingertip points and the valley points between adjacent fingers, are extracted by travelling along the hand boundary and searching for curvature extremities and corner point detection.



**Fig. 2.** Extracted points in human hand image



**Fig. 3.** Areas of human hand regions

'Corner' or 'dominant point' detection is important for pattern or picture analysis. Corners are important image hand features since they correspond to unique features of hand and are invariant to many transformations.

A corner is (informally) defined as a high-curvature point on a simple digital arc or curve. Corners can be used to segment arcs or curves. We consider curves  $\rho$  in the digital plane. Pixels  $p_i$  in such a digital curve  $\rho = p_0, p_1, \dots, p_{n-1}$  have coordinates  $(x_i, y_i)$ . In order to detect a corner at the pixel  $p_i$  on a curve  $\rho$ , it is common practice that a corner detector considers an angular measure based on a predecessor  $p_{i-b}$ ,  $p_i$  itself, and a successor  $p_{i+f}$ , where  $b, f > 0$  are fixed or variables within a defined interval.

We assume that hand boundary curve is estimated from an edge detector. A corner point on such a curve subtends a sharp angle between its neighbouring points. A discrete boundary curve is described by  $r_k = [x_k, y_k]^t$ ,  $k = 0, 1, \dots, n - 1$ , where  $t$  is the matrix transpose, and  $x_k$  and  $y_k$  denote the  $x$  and  $y$  coordinates in the 2D image plane. In the first step, the curve is smoothed to prevent variations in the estimated angular values:

$$\tilde{x}_k = \frac{1}{2w + 1} \sum_{l=k-w}^{k+w} x_l, \quad \tilde{y}_k = \frac{1}{2w + 1} \sum_{l=k-w}^{k+w} y_l, \tag{1}$$

where  $\widetilde{(\cdot)}$  is smoothed value of  $(\cdot)$  and  $w$  is a small integer number.

The angle  $\varphi_k$ , associated with each curve point  $r_k$ , is then computed as the angle between the two vectors  $(\tilde{r}_{k+1}, \tilde{r}_k)$  and  $(\tilde{r}_k, \tilde{r}_{k-1})$  using the smoothed curve points, such as:

$$\varphi_k = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right), \tag{2}$$

where  $a = |\tilde{r}_{k-1} - \tilde{r}_k|$ ,  $b = |\tilde{r}_{k+1} - \tilde{r}_k|$  and  $c = |\tilde{r}_{k+1} - \tilde{r}_{k-1}|$ .

Significant corners are extracted from those points which have the local minimum angular values.

Four points of them are the top points of the index, middle, ring and little fingers. The other three points are the joining points - one point is that between index and middle, the second one is between middle and ring and the last one is between ring and little fingers. We also extracted two additional points. These points are marked in Figure 2.

The final nine points are used for the feature measurement. Using these nine points, features for hand geometry verification are extracted. Eight distinct triangles can be formed using the nine points. We calculate the eight areas of hand finger regions (Figure 2 and Figure 4).

The hand descriptor is represented by a feature vector. Thus the feature vector  $V$  has a size of  $N$ .

$$VH = Y_1, Y_2, \dots, Y_N, \tag{3}$$

where  $Y_i$  represents the feature and in our case  $N = 8$ .

The distance,  $D$ , between two feature vectors of two hand images is calculated using:

$$D = \frac{1}{N} \sqrt{\sum_{i=1}^N \left(\frac{Y_i^j - Y_i^k}{\mu_i}\right)^2}, \tag{4}$$

where  $Y_i^j$  and  $Y_i^k$  are the  $i$ th of the feature vectors of hand  $j$  and hand  $k$ , and:

$$\mu_i = \frac{Y_i^j + Y_i^k}{2}. \tag{5}$$

### 3.2 Palmprint Feature Extraction

Palmprint, which is one of the physiological biometric features, is a perfect biometric identifier because of its stability and uniqueness. The rich texture feature information of human palmprint places palmprint as one of the powerful means in personal identification and authentication. Palmprint verification system is one of the most interesting biometrics approaches which offers significant advantages; it is non-intrusive, user-friendly, requires low spatial resolution imaging and has stable as well as unique features [17].

It is very difficult, if not impossible, to use one feature model for palmprint matching with high performance in terms of accuracy, efficiency, and robustness. Palmprints are stable and show high accuracy in representing the identity of each individual [14]. Thus, they have been commonly used in law enforcement and forensic environments.

The extracted palmprint images are normalized to have pre-specified mean and variance. The normalization is used to reduce the possible imperfections in the image due to sensor noise and non-uniform illumination.

Palmprint features can be divided into three different categories: a) point features, which include minutiae features from ridges existing in the palm, and delta point features, from delta regions found in the finger-root region; b) line features, which include the three relevant palmprint principal lines, due to flexing the hand and wrist in the palm, and other wrinkle lines and curves (thin and irregular); and, c) texture features of the skin.

Since lines are the dominant features to describe palmprint [4], we implement shape feature extraction technique, namely Zernike Moment Invariants, in the application of human palmprint authentication.

The Zernike moments [11, 18] are built using a set of polynomials that form a complete orthogonal base defined in the unit circle  $(x^2 + y^2) \leq 1$  (Figure ??). These moments are projections of the input image,  $f(x, y)$  in the space of the orthogonal functions:

$$V_{n,m}(x, y) = R_{n,m}(x, y) \cdot e^{j m \operatorname{atan}(\frac{y}{x})}, \tag{6}$$

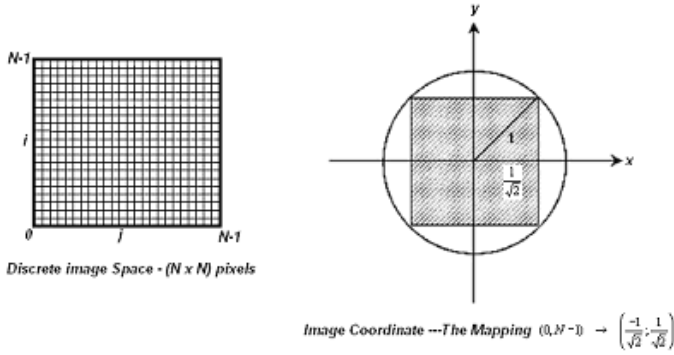
where  $j = \sqrt{-1}$ ,  $n \geq 0$ ,  $|m| \leq n$ ,  $n - |m|$  is odd, and

$$R_{n,m}(x, y) = \sum_{s=0}^{\frac{n-|m|}{2}} \frac{(-1)^s (x^2 + y^2)^{\frac{n}{2}-s} (n-s)!}{s!((n + \frac{|m|}{2}) - s)!((n - \frac{|m|}{2}) - s)!}. \tag{7}$$

For a discrete image, the Zernike moments of order  $n$  and repetition  $m$  are given by:

$$A_{n,m} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) \cdot V_{n,m}^*(x, y), \tag{8}$$

where  $(x^2 + y^2) \leq 1$ ,  $n \geq 0$ ,  $|m| \leq n$ ,  $n - |m|$  is odd and the symbol  $*$  represents the conjugated complex operator. The magnitudes  $|A_{n,m}|$  are invariable to the rotation  $A_{n,m} = A_{n,-m}$ .



**Fig. 4.** The square to circle transformation

Equation (8) can be also written as follows:

$$A_{n,m} = \frac{n+1}{\pi} \sum_{\rho} \sum_{\theta} f(\rho \cos \theta, \rho \sin \theta) \cdot R_{n,m}(\rho) \cdot e^{jm\theta} \text{ for } \rho \leq 1. \quad (9)$$

The image reconstructed from Zernike moments up to given moment  $N$  is defined as follows:

$$I_N(x, y) = \sum_{n=0}^N \sum_m A_{n,m} \cdot V_{n,m}(x, y), \quad (10)$$

where  $(x^2 + y^2) \leq 1$ , and  $m$  have similar constrains as in eq. (6).

The original image can be obtained as:

$$f(x, y) = \lim_{n \rightarrow \infty} I_N(x, y). \quad (11)$$

The magnitude of Zernike moments has rotational invariance property. An image can be better described by a small set of its Zernike moments than any other type of moments.

To characterize the palmprint we used a feature vector:

$$ZMI = (A_{1,m}, A_{2,m}, \dots, A_{n,m}) \quad (12)$$

consisting of the calculated Zernike moments.

This vector is used to index each palmprint in the database. The distance between two feature vectors is determined by city block distance measure.

## 4 Fusion, Experiments and Results

An input image is converted into a set of features, and then is matched with the claimant's hand image stored in the database and one distance metrics are applied to calculate the similarity between the two feature vectors.

In our biometric system the fusion is performed at the matching-score level. When trying to verify the identity of an unknown sample we receive two sets of scores from the two independent matching modules:

- (i)  $D(VH_x, VH_j)$ , where  $VH_x$  is the unknown hand-template feature vector, and  $VH_j$ ,  $j = 1, 2, \dots, n$  are the hand-templates vectors stored in the database under the identity the system is trying to verify;
- (ii) Similarity measures  $Q(ZMI_x, ZMI_j)$  where  $ZMI_x$  is the unknown palmprint-template feature vector, and  $ZMI_j$ ,  $j = 1, 2, \dots, n$  are palmprint-templates feature vectors stored in the database under the identity the system is trying to verify.

In order to generate the unique matching score we need a way to combine individual matching scores from hand and palmprint-matching modules.

Since the palmprint-matching scores and the hand-matching scores come in different ranges, a normalization has to be performed before they are combined. The normalization is carried out by means of two transition functions, which map the distances  $D$ , and similarity measures  $Q$ , into the interval  $[0,1]$  were determined experimentally from the training set of the database.

We have experimented the whole hand recognition system on database of 100 subjects. For each of them, three images are catalogued. On the hand image dataset, the multimodal system correctly recognized 274 hands. The testing results are listed in Table 1. The Rank-1 recognition rate is 90.33% which proves that the proposed approach is effective in hand-palm identification.

**Table 1.** Rank-1 recognition results

Biometrics modality	Number of hand images	Positive detected hands	Rank-1 rate
Hand only	300	241	80,33%
Palm only	300	258	86,00%
<b>Hand+Palm</b>	300	274	<b>91,33%</b>

## 5 Conclusions

In this work we developed a new approach to hand geometry feature extraction by using curvature analysis. Moreover, we proposed palmprint feature extraction method based on Zernike Moments. Then we performed fusion on a feature level and integrated both modalities into a hybrid one-sensor biometrics for human identification. The objective of this work was to investigate the integration of palmprint and hand geometry features, and to achieve higher performance that may not be possible with single biometric indicator alone. We have achieved satisfactory results and proved that hand-palm multimodal biometrics may be applied in medium-security applications.

## References

1. Khotanzad, A. Invariant image recognition by Zernike moments. *IEEE Trans. Pattern. Anal. Machine Intell.*, 12:489–497, May 1990.
2. Mukundan, R., Ramakrishnan, K.R. *Moment Functions in Image Analysis Theory and Applications*. World Scientific Publishing, 1998.
3. Sanchez-Reillo R., Sanchez-Avila C., Gonzales-Marcos A. Biometric Identification through Hand Geometry Measurements. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 22(10), 1168-1171, 2000.
4. Jain, A.K., Ross, A., Pankarti, S. A prototype hand geometry based verification system. *Proc. AVBPA*, Washington D. C.:166–171, March 1999.
5. Bulatov, Y., Jambawalikar, S., Kumar, P., Sethia, S. *DIMACS Workshop on Computational Geometry*, chapter Hand recognition using geometric classifiers. Rutgers University, Piscataway, NJ, 2002.
6. Oden, C., Ercil, A., Buke, B. Combining implicit polynomials and geometric features for hand recognition. *Pattern Recognition Letters*, 24:2145–2152, 2003.
7. Lay, Y.L. Hand shape recognition. *Optics and Laser Technology*, 32(1):1–5, Feb. 2000.
8. Saeed K., Werdoni M. A New Approach for Hand-Palm Recognition. In: Pejaś J. and Piegat A. (Eds), *Enhancement Methods in Computer Security Biometric and Artificial Intelligence Systems*, New York, USA, 2005, Springer Science + Business Media.
9. Han C.C., Cheng H.L., Lin C.L., Fan K.C. Personal Authentication using Palmprint Features. *Pattern Recognition*, 36:371-381,2003.
10. Kumar A, Wong D.C.M., Shen H.C., Jain A.K. Personal Verification using Palmprint and Hand Geometry Biometric. *Proc. AVBPA*, 2003, 668-678.
11. Kumar A., Shen H.C. Recognition of Palmprints using Wavelet-based Features. *Proc. of Intl. Conf. on Systems and Cybernetics*, 2002.
12. Li W., Zhang D., Xu Z. Palmprint Recognition by Fourier Transform. *Journal of Software*, 13(5), 879-886, 2002.
13. Li W., Xia S., Zhang D., Xu Z. A New Palmprint Segmentation Method Based on an Inscribed Circle. *Image Processing and Communications*, 9(1), 63-70,2002.
14. You J., Li W., Zhang D. Hierarchical Palmprint Identification via Multiple Feature Extraction. *Pattern Recognition*, 35, 847-859, 2002.
15. Zhang D., Shu W. Two Novel Characteristics in Palmprint Verification: Datum Point Invariance and Line Feature Matching. *Pattern Recognition*, 33(4), 691-702, 1999.
16. Canny, J.F. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, June 1986.
17. Chen, J., Zhang, C., Rong, G. Palmprint recognition using crease. *Proc. Intl. Conf. Image Process.*, 234–237, Oct. 2001.
18. Liao, S.X., Pawlak, M. On the accuracy of zemike moments for image analysis. *IEEE Trans. Pattern. Anal. Machine Intell.*, 20:1358–1364, Dec. 1998.

# A Study on Iris Feature Watermarking on Face Data

Kang Ryoung Park<sup>1</sup>, Dae Sik Jeong<sup>2</sup>, Byung Jun Kang<sup>2</sup>, and Eui Chul Lee<sup>2</sup>

<sup>1</sup> Division of Digital Media Technology, Sangmyung University,  
7 Hongji-Dong, Jongro-Gu, Seoul, Korea  
Biometrics Engineering Research Center (BERC)  
parkgr@smu.ac.kr

<sup>2</sup> Department of Computer Science, Sangmyung University,  
7 Hongji-Dong, Jongro-ku, Seoul, Republic of Korea  
Biometrics Engineering Research Center (BERC)

**Abstract.** In this paper, we propose a new iris feature watermarking method on face data. This research has following three objectives. First, by using watermarked iris features in addition to face data, the multimodal biometric authentication can be possible, which can increase the authentication accuracy. Second, in case that the saved face data is illegally let out and privacy infringement happens, by checking the inserted iris feature watermark, we can solve the legal responsibility problem about the outflow of face data. In detail, if the iris feature watermark cannot be extracted from the outflow face data, we can insist that the face data is let out from other organization instead of ours. Third, in case that “the iris features need to be transmitted via non-secure and noisy communication channel” [1], it can be invisibly hidden on face data by our method. For the first objective, the face recognition accuracy with iris feature watermark should not be degraded. For the second and third objectives, the inserted iris watermark should be “strong” enough to be extracted irrespective of various kinds of attacks (such as blurring, cropping and rotation attacks) and noise insertion on face data. This research has three advantages compared to previous works. First, to overcome the vulnerability of blurring attack to previous biometric watermarking based on spatial domain, we use the watermarking method in frequency domain. Second, to reduce the degradation of face recognition accuracy due to iris watermarking, we insert the watermark into mid and high frequency bands. Third, through using individual unique iris features for biometric watermarking information and secondary authentication, the security level is much enhanced and we can solve legal responsibility problem about the outflow of face data. Experimental results showed that our algorithm could be used to accomplish above objectives.

## 1 Introduction

Biometrics is the science of recognizing a person's identity by using human physiological or behavioral characteristics. With the wide spread of biometric authentication technology in many applications, it is often the case that “the biometric features need to be transmitted via non-secure and noisy communication channel” [1]. In such a case, the issue to protect the biometric features to be transmitted is raised. In addition,



to increase the authentication accuracy, it is required to combine more than two biometrics. So, the technology of combining biometrics and watermarking algorithm has been proposed [1][2][3]. Watermarking technology was originally introduced in order to hide some invisible information into image or audio data. Based on that, it can be used to protect the ownership of the watermarked data or to guarantee the integrity of the watermarked data. For the former objective, “strong watermarking” is adopted and “weak watermarking” is used for the latter objective.

Jain et al.[1][2] have proposed the method of hiding fourteen eigen-face coefficients in fingerprint images based on digital watermarking technique in order to protect the face features and increase the authentication accuracy by both face and fingerprint recognition. Although a supervisor is able to identify an extracted eigen-face, there are some further disadvantages. First, fingerprint images could be easily impaired by embedding face watermarks because the local features such as minutiae are used for conventional fingerprint recognition. Second, their and Minerva et al.[3]’s method for extracting watermark use local-based 5x5 cross-shaped neighborhood, but such a method based on spatial domain has severe weakness against attack such as blurring. In detail, in case of making fingerprint image blurred a little, the watermarked face coefficients could not be extracted. In another researches [4], Jaehyuck Lim et al proposed the invertible watermarking algorithm based on compressed RS (Regular and Singular) bit streams. Though manipulation positions of watermarked biometric data can be detected, their method has disadvantage that it has weakness to various kind of attacks such as blurring and cropping.

In terms of biometric watermarking, iris image is also sensitive to watermark embedding, because it uses high frequency components such as fine iris textures for authentication. On the contrary, a face image has the advantage of being embedded with a more robust watermark, because it uses low frequency components for authentication compared to iris and fingerprint image. However, face recognition has an inherent weakness for high security authentication due to its sensibility to pose, expression and lighting variations. So, we propose a new method of watermarking iris features on face data. For that, we hide iris features based on FFT (Fast Fourier Transform)-based watermarking technique in a face image.

## 2 Proposed Biometric Watermarking Method

### 2.1 Overview of Proposed Method

In this research, we use iris features as watermarking information inserted to face image. After face recognition is finished, we extract the inserted watermark (iris features in frequency domain) from face image and compute the similarity between the extracted iris features and the enrolled one. This is not only a robust authentication method, but it can make the watermarked iris features secure. In addition, in case that face data is illegally let out, we can solve the legal responsibility problem by checking watermarked iris features. This is the proposed method that we used as shown in Fig.1.

### < Watermark Encoding Part >

1. *Face Image Acquisition*: Face image is captured. Then, face region is located by Adaboost face detector [5] and eye locations are also detected [6].
2. *Normalization*: Normalizing the size of face image into 30×30 pixels based on the detected eye positions.
3. *Extracting Iris Features*: From the input iris image, iris region is segmented including removing eyelash, eyelid and SR detection as shown in Fig.2. Then, we extract iris features in mid & high frequency bands of FFT domain from the segmented iris region.
4. *Iris Feature Watermark Inserting*: Inserting extracted iris features into normalized face images in FFT domain as digital watermark. In detail, we add the frequency amplitudes of iris features to those of face image.

### < Watermark Decoding and Authentication Part >

5. *Face Matching*: The watermarked face image is transformed into eigen-face by conventional PCA (Principal Component Analysis) algorithm [7]. Then, we compute the similarity between the enrolled and the acquired eigen-coefficients by Euclidian distance. The threshold for authentic or imposter was determined by trained face data (without watermark) based on Bayesian rule which can minimize both FAR (False Acceptance Rate: error rate of accepting imposter as genuine) and FRR (False Rejection Rate: error rate of rejecting genuine as imposter).
6. *Iris Feature Watermark Extracting and Iris Matching*: If the input face image is matched with the enrolled one, we extract the iris feature watermark in face image. For that, we calculate the correlation value in FFT domain between the face image (including iris feature watermark) and enrolled iris features as shown in Fig.3. Iris authentication is successful when the computed correlation value is greater than predefined threshold. If the correlation value is smaller than the threshold, iris authentication is regarded to be failed.

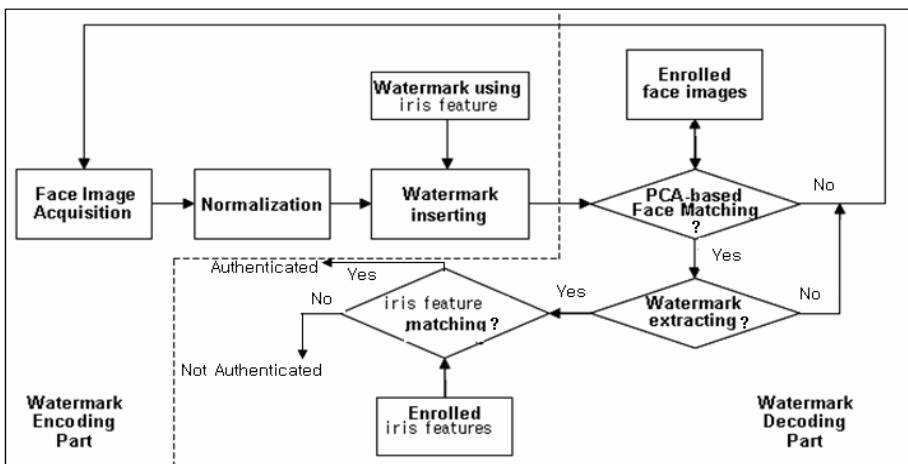
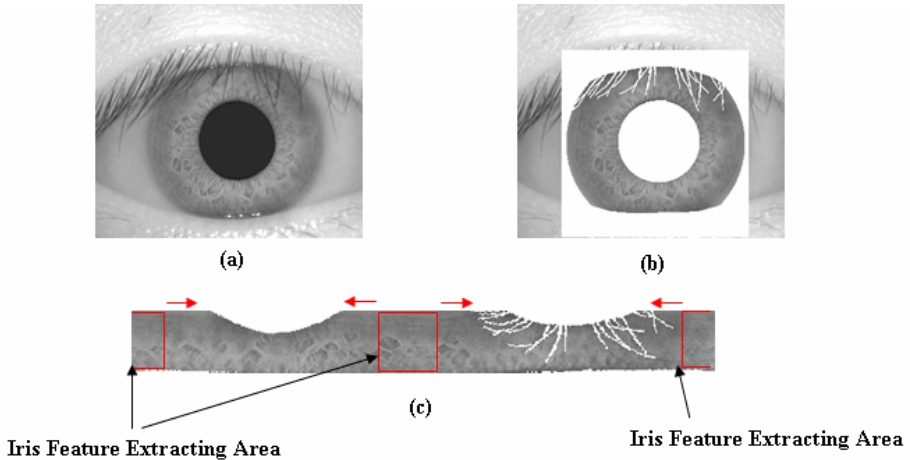


Fig. 1. The overview of proposed method

## 2.2 Inserting and Extracting Iris Features Watermark

In this section, we explain the method of inserting and extracting iris features watermark. When eye image is captured, in order to isolate iris region from the eye image, we perform pupil and iris detection based on circular edge detection method [8][9]. Upper and lower eyelids are also located by eyelid detection mask and parabolic eyelid detection method [8][10]. Then, we determine the eyelash candidate region based on detected iris & pupil area and detect the eyelash region [11][12] as shown in Fig. 2(b).



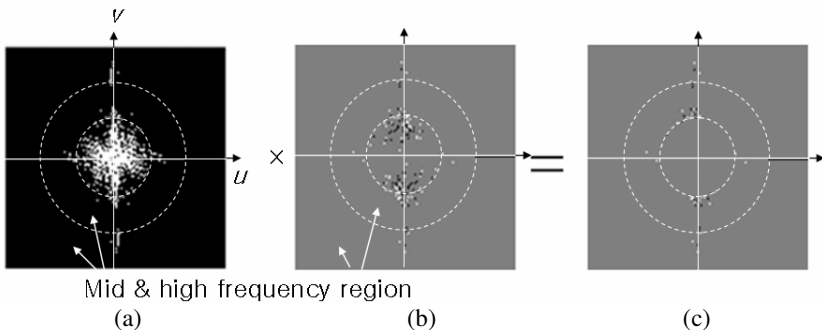
**Fig. 2.** Iris region segmentation and normalized iris image. (a) Original Image (b) Detected iris region (c) Normalized image.

After that, the detected circular iris region is normalized as rectangular shape as shown in Fig. 2(c). In general, each iris image has variations about the length of its iris outer boundary. That is because there exists the size variation of iris per user (it is reported that the diameter of iris is about 10.7 ~ 13mm). Another is because the captured image size of iris may be changed according to Z distance between camera and eye. So, we adjust the length of iris outer boundary into 256 pixels by stretching and interpolation as shown in Fig. 2(c). Then, we define iris feature extracting area which does not include eyelid, eyelash and SR (Specular Reflection). We determine the region of which gray level is above 250 as SR) as shown in Fig.2(c). Sometimes, the some part of eyelash, eyelid or SR can be included in the iris feature extracting area. In such a case, we make the occluded part be zero-padded, which is not used for FFT. In addition, we move the iris feature extracting area horizontally in order to cope with the eye rotation as shown in Fig.2(c). In this case, we permit the eye rotation of -10 ~ +10 degrees. Then we transform the iris feature extracting area by FFT and extract the components in mid & high frequency band, which are used for the watermark to be inserted. The reason why we use the components in mid & high frequency band for iris watermarking features is that the fine texture of iris pattern (which is used for iris recognition) is mainly shown as “mid & high” frequency component. Though the

amount of low frequency component such as facial skin, pupil and sclera (as shown in Fig. 2(a)) is greater than that of the mid & high frequency ones as shown in Fig. 3(b), they play little role on iris recognition. Whereas, it is reported that the majority of facial eigen-features (which play important role in recognition) and most information of face image exist in “low” frequency band [19]. To insert the iris feature watermark, we add the extracted iris feature components (amplitude) in mid & high frequency band to the facial feature components (amplitude) in FFT domain. In such case, the added amplitude of iris feature components in mid & high frequency band can be controlled by our watermarking strength parameter.

To extract the embedded watermark, we compute the correlation (Corr) in FFT domain between the watermarked face image and the iris feature extracting area (Fig.2) of the enrolled iris image. If the correlation value (Corr) exceeds in the threshold, we determine that the watermarked iris features is matched with the enrolled one. The threshold for authentic or imposter was determined by trained iris data based on Bayesian rule which can minimize both FAR and FRR.

Figure 3 shows watermark extracting process. Each image of Fig.3 shows the power spectrum in FFT domain. The horizontal and vertical axis of each image represents the horizontal and vertical frequency value of  $u$  and  $v$ , respectively. In each image, low frequency component is shown in the center of image and the higher frequency components are shown in outer region from image center such as conventional FFT image. In this case, because we calculate correlation (Corr) in wide range of area such as mid & high frequency region, our method is more robust against blurring attack than Jain’s method [1][2] using local-based  $5 \times 5$  cross-shaped neighborhoods in spatial domain. In detail, we multiply the power spectrums in mid & high frequency region of watermarked face image (Fig.3 (a)) with those of enrolled iris features (Fig.3 (b)). From that, we use the calculated sum of multiplication values as Corr. In this case, because we only use the FFT power spectrum for iris features and watermarks without phase information, the spatial position information of iris features and watermarks are not known. So, even if an attacker steals the iris features or watermarks, he cannot recover original iris information from them.



**Fig. 3.** Extracting iris feature watermark using correlation in FFT domain(a) Watermarked face image in frequency domain (b) Enrolled iris features in frequency domain (c) Correlation result and extracted iris feature watermark in frequency domain

### 3 Experimental Results

To prove the validity of our approach and check whether our algorithm accomplishes the research objectives mentioned in abstract, we tested with two biometric databases. The first database is the BioID Face DB [13]. This dataset consists of 1,521 face images (8 bit gray-level) with a resolution of 384x286 pixels. Each image has the frontal view of a face for 23 subjects. The second dataset is the CASIA DB [14] for the iris dataset. The dataset consists of 756 iris images (108 eyes from 80 subjects) having 8-bit gray-level with 320x280 pixels. To evaluate the performance of the proposed methods, we have performed two experiments. In the first experiment, we tested the change of face recognition accuracy due to the embedding of iris watermarking. The accuracy was measured based on EER (Equal Error Rate) according to the watermark strength.

**Table 1.** The change of face recognition accuracy due to the embedding of iris watermarking

Watermarking Strength	0	10	20	30	40	50	60	70
EER (%)	12	12	12.1	12.1	12.9	13.1	13.2	13.5

In table 1, strength 0 means no watermarking and strength 70 does maximum watermarking. When we use the strength below 30, the EERs were almost same and we can know the watermarked face image can be used for face authentication without accuracy degradation. Even in case of maximum strength 70, EER was increased only about 1.5% compared to no watermarking. In all cases, the authentic distribution was moved to the imposter's one according to watermarking strength, whereas the imposter one was not moved. In the second experiment, the accuracy (EER) of iris recognition with the extracted iris features from face image was measured. As shown in Table 2 and 3, the EER were measured according to blurring and cropping attacks. In this case, the accuracy (EER) of iris recognition with the watermarked iris features represents the robustness of the inserted watermark against various attacks. In all cases, the watermarking strength was 30 by considering face authentication accuracy as mentioned in the first experiment.

**Table 2.** The accuracy (EER) of iris recognition with the watermarked iris features (with blurring attack by Gaussian Blurring)

Radius of Gaussian Blurring Kernel (pixel)	No-blurring	1	2	3	4	5	6
EER (%)	0.10	0.11	0.12	0.12	0.12	0.12	0.12

As shown in Table 2, the EER which was measured between the watermarked iris features and the enrolled one is not changed much irrespective of blurring attacks. That is because the correlation in mid frequency range is used in addition to that in high frequency range for iris recognition. For attack, we made the watermarked face image blurred by Gaussian blurring kernel. Comparing the performance of our method to that

of Jain et al.[1][2], their method could not extract the inserted face watermark even in case that the radius of Gaussian blurring is more than 2 pixels. From that, we can know that our algorithm is superior to their method about blurring attack.

**Table 3.** The accuracy (EER) of iris recognition with the watermarked iris features (with cropping attack)

The Ratio of Cropped Area to Whole Face Region (%)	No-Cropping	5	10	15	20	25	30
EER (%)	0.10	0.11	0.11	0.12	0.12	0.12	0.12

As shown in Table 3, the EER which was measured between the watermarked iris features and the enrolled one is not changed much irrespective of cropping attacks. That is because the watermark of iris features are inserted in whole face region and local cropping attack can not make much degradation of accuracy. In the next attacking test, we measured the EER according to the rotation angle of face image.

**Table 4.** The accuracy (EER) of iris recognition with the watermarked iris features (with rotation attack)

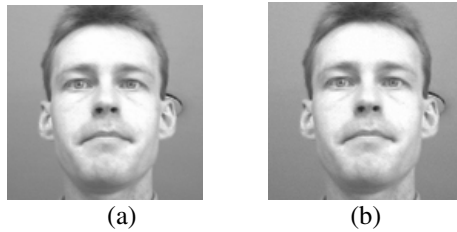
The rotation angle (degree)	0	20	40	60	80
EER (%)	0.1	0.1	0.1	0.11	0.1

From Table 4, we can know that performance was not affected by the rotation attack. In the next attacking test, we also measured the EER according to the amount of inserted Gaussian noise on face image. The amount of inserted noise was defined as SNR (Signal to Noise Rate,  $SNR = 10 \times \log_{10} (P_s/P_n)$ , where  $P_s$  is the variance of original image and  $P_n$  is that of noise image).

**Table 5.** The accuracy (EER) of iris recognition with the watermarked iris features (with the attack of inserting Gaussian noise)

SNR (dB)	0	30	20	10	5
EER (%)	0.1	0.1	0.1	0.23	0.28

From Table 5, we can know that the EER was increased in case that the SNR is lower than 20 dB. That is because much high frequency component of noise was inserted in face image, but in such case, the noise images of face image could be perceived by administrator. Most cases of Table 2 ~ 5, our accuracies (EER) of iris recognition were superior to that of previous iris recognition method [15] (In [15], the accuracy (EER) was 0.275 ~ 0.361%) irrespective of watermarking and various kinds of attacks. Fig.4 shows the original image and the watermarked face image with the strength of 30. As shown in Fig.4, the watermarked face image is seen as same to the original face image.



**Fig. 4.** The results of original and watermarked face image (a) Original image (b) Watermarked face image with strength 30

## 4 Conclusions and Future Work

In this paper, we propose a new iris feature watermarking method on face data having following three objectives. First, by using watermarked iris features in addition to face data, the multimodal biometric authentication can be possible, which can increase the authentication accuracy. Second, in case that the saved face data is illegally let out and privacy infringement happens, by checking the inserted iris feature watermark, we can solve the legal responsibility problem about the outflow of face data. Third, in case that “the iris features need to be transmitted via non-secure and noisy communication channel” [1], it can be invisibly hidden on face data by our method. In our future work, we will have more tests with various kinds of attacks such as distortion. In addition, in case that many specular reflections happen in iris region by glasses surface, the performance of iris authentication is degraded, which should be solved by future work. And, by using wavelet transform, we plan to study the watermark inserting method in both spatial and frequency domain in future work.

**Acknowledgements.** This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center (BERC) at Yonsei University.

## References

1. Lin H. and Anil. K.J.: Integrating Faces and Fingerprints for Personal Identification. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 12, pp. 1295-1307, 1998
2. Anil K.J., Umut U. and Rein-Lien H.: Hiding a Face in a Fingerprint Image. Proc. of Int. Conf. on Pattern Recognition. vol. 3, pp. 756-759, 2002
3. Minerva M. and Sharath P.: Verification Watermarks on Fingerprint Recognition and Retrieval. Proc. of SPIE Conference on Security and Watermarking of Multimedia Contents, vol. 3657, 1999
4. Jaehyuck Lim *et al.*: Invertible Watermarking Algorithm with Detecting Locations of Malicious Manipulation for Biometric Image Acquisition, LNCS, vol. 3832, pp. 763-769, 2006
5. Z. Ou, : Cascade AdaBoost Classifiers with Stage Optimization for Face Detection, LNCS, vol. 3832, pp. 121-128, 2006

6. Guo C.F. and Pong C.Y.: Multi-cues eye detection on gray intensity image. *Journal of the Pattern Recognition*, vol. 34, pp. 1033-1046, 2001
7. Turk, M. and Pentland, A.: Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*. vol. 3, pp. 71-86, 1991
8. John G. Daugman: How Iris Recognition Works. *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21-30, 2004
9. Dal-ho Cho *et al.*, : Pupil and Iris Localization for Iris Recognition in Mobile Phones, SNPD, Las Vegas Nevada, USA, June 19-20, 2006
10. Y.K Jang *et al.*, Robust Eyelid Detection for Iris Recognition, *Journal of IEEK*, submitted
11. Byung Joon Kang, Kang ROUNG Park : A Study on Iris Image Restoration, *Lecture Notes in Computer Science on AVBPA 2005*, vol. 3546, pp.31-40, 2005
12. Byung Jun Kang *et al.*, : A Robust Eyelash Detection Based on Iris Focus Assessment, *Pattern Recognition Letters*, submitted
13. BioID Face DB of Humanscan Corp., <http://humanscan.de/support/downloads/facedb.php> (accessed on 14 December 2006)
14. CASIA iris DB of Chinese Academy of Science, <http://www.sinobiometrics.com/resources.htm> (accessed on 14 December 2006)
15. Seung-In Noh, Kwanghyuk Bae, Kang Ryoung Park, and Jaihie Kim : A New Feature Extraction Method Using the ICA Filters for Iris Recognition System, *Lecture Notes in Computer Science (IWBRIS'05)*, vol. 3781, pp. 142-149, 2005



# Keystroke Dynamics for Biometrics Identification

Michał Choraś<sup>1</sup> and Piotr Mroczkowski<sup>2</sup>

<sup>1</sup> Image Processing Group, Institute of Telecommunications,  
University of Technology & Life Sciences,  
S. Kaliskiego 7, 85-791 Bydgoszcz, Poland  
chorasm@atr.bydgoszcz.pl

<sup>2</sup> Hewlett Packard Polska, Global Delivery Poland Center,  
ul. Szturmowa 2a, University Business Center, Warsaw, Poland  
piotr.mroczkowski@hp.com

**Abstract.** Personal identification has lately become a very important issue in the still evolving network society. Biometrics identification methods proved to be very efficient, more natural and easy for users than traditional methods of human identification. Hereby we discuss the idea of human identification based on keystroke dynamics. In the article we focus on our methods of feature extraction from the typing patterns. Moreover, we present satisfactory experimental results and possible applications of keystroke biometrics.

## 1 Introduction

There are many known methods of human identification based on biometrics characteristics. In general, these biometrics methods can be divided into behavioral and physiological regarding the source of data. The first class is based on the behavioral features of human actions and it identifies people by how they perform something.

**Keystroke dynamics** biometric systems analyze the way a user types at a terminal by monitoring the keyboard events, and thus is considered as the behavioral approach. Identification is based on the rhythm of typing patterns, which is considered to be a good sign of identity [1], [2], [3]. In other words not what you type, but how you type is important. In this approach several things can be analyzed: time between key-pressed and key-released events, break between two different keystrokes, duration for digraphs and trigraphs and many more.

Keystroke verification techniques can be divided into two categories: static and continuous. Static verification approaches analyze keyboard dynamics only at specific times, for example during the logon process. Static techniques are considered as providing a higher level of security than a simple password-based verification system [1]. The main drawback of such an approach is the lack of continuous monitoring, which could detect a substitution of the user after the initial verification. Nevertheless, the combination of the static approach with password authentication was proposed in several papers [4] and it is considered as

being able to provide a sufficient level of security for the majority of applications. The human identification keystroke biometrics system proposed here is based on such a combination.

Continuous verification, on the contrary, monitors the user's typing behavior through the whole period of interaction [1]. It means that even after a successful login, the typing patterns of a person are constantly analyzed and when they do not match user's profile access is blocked. This method is obviously more reliable but, on the other hand, the verification algorithms as well as the implementation process itself are much more complex.

## 2 Related Work

One of the first studies on keyboard biometrics was carried out by Gaines et al. [5]. Seven secretaries took part in the experiment in which they were asked to retype the same three paragraphs on two different occasions in a period of four months. Keystroke latency timings were collected and analyzed for a limited number of digraphs and observations were based on those digraph values that occurred more than 10 times [6].

Similar experiments were performed by Leggett with 17 programmers [4]. In the 15 last years, much research on keystroke analysis has been done (e.g., Joyce and Gupta [7], Bleha et al. [8], Leggett et al. [4], Brown and Rogers [9], Bergadano et al. [10], and Monroe and Rubin [1], [6]).

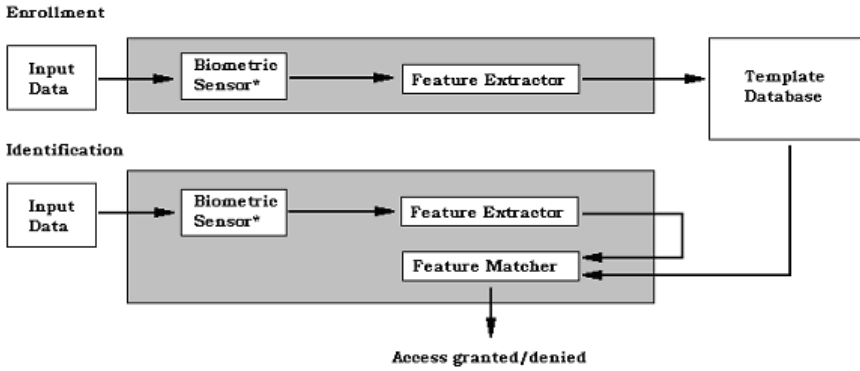
Several proposed solutions got U.S. patents (for instance Brown and Rogers [11]). Some neural network approaches (e.g., Yu and Cho [12]) have also been undertaken in the last few years. More recently, several papers where keystroke biometrics, in conjunction with the login-id password pair access control technique, were proposed (e.g., Tapiador and Sigenza [13]). Some commercial implementations are also available ('Biopassword', a software tool for Windows platform commercialized by Net Nanny Inc. [14]).

## 3 Feature Extraction in Biometrics Based on Keystroke Dynamics

In every biometric identification system developing appropriate methods of feature extraction is the major problem to be solved. Biometric features should be unique, collectable, measurable and accepted by the users.

In the following section we present features that are extracted from keystroke characteristics in our system.

In the implemented verification system three independent methods of the identity verification are performed every time a user attempts to log in. The first method compares typing paths stored in the database against a typing path created at the time of logon process. Two other methods are based on the calculation of the degree of disorder of digraphs and trigraphs respectively.



**Fig. 1.** An overview of a biometric system [15]. \*In case of keyboard dynamics no real sensor is needed - characteristics are measured using timer.

### 3.1 Typing Paths

Typing paths can be described as a set of key code/key event pairs stored in order of occurrence. If some short sequence of chars is being retyped by a user several times (which is the case with the "Login - Password" mode), the analysis of such paths is likely to show some typical characteristics of a user's behavior:

- moments where keys overlap (second key is pressed before the release of the first one)
- the position of the key pressed in the case of duplicate keys (digits, SHIFT's, etc.)

### 3.2 Digraphs and Trigraphs

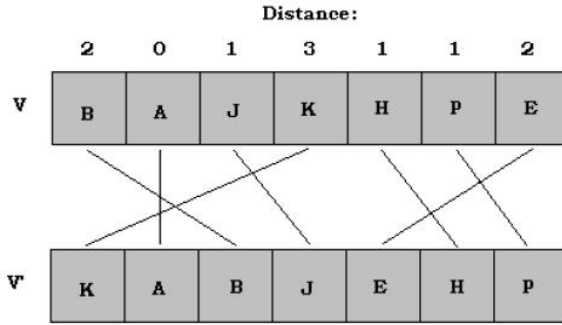
Digraph is defined as two keys typed one after the other. In our case the duration of a digraph is measured between the press event of the first key and release event of the second key.

Trigraph is defined as three keys typed one after the other. The duration of trigraph is measured between pressing event of the first key and release of the third key.

### 3.3 Degree of Disorder

Having two sets of key latencies of the same Login-Password pair, it is possible to measure their "similarity". One way to calculate that is the degree of disorder (*do*) technique [10].

Let us define vector  $V$  of  $N$  elements and vector  $V'$ , which includes the same  $N$  elements, but ordered in a different way. The degree of disorder in vector  $V$  can be defined as the sum of the distances between the position of each element in  $V$  with respect to its counterpart vector  $V'$ . If all the elements in both vectors are in the same position, the disorder equals 0.



**Fig. 2.** The distances between the position of each element in  $V$  with respect to  $V'$

Maximum disorder occurs when elements in vector  $V$  are in the reverse order to the model vector  $V'$ .

Maximum disorder ( $do_{max}$ ) is given by:

$$do_{max} = \begin{cases} \frac{|V|^2}{2} & \text{if } |V| \text{ is even} \\ \frac{(|V|^2-1)}{2} & \text{if } |V| \text{ is odd} \end{cases} \tag{1}$$

where  $|V|$  is length of  $V$ .

In order to get the normalized degree of disorder ( $do_{nor}$ ) of a vector of  $N$  elements, we divide  $do$  by the value of the maximum disorder. After normalization, the degree of disorder falls between 0 ( $V$  and  $V'$  have the same order) and 1 ( $V$  is in reverse order to  $V'$ ). For the vector  $V$  in Figure 2 the disorder can be calculated as:

$$do = (2 + 0 + 1 + 3 + 1 + 1 + 2) = 10, \tag{2}$$

where  $do_{max}$  equals:

$$do_{max} = \frac{(|V|^2 - 1)}{2} = \frac{7^2 - 1}{2} = \frac{48}{2} = 24. \tag{3}$$

In order to normalize the disorder, we perform:

$$do_{nor} = \frac{do}{do_{max}} = \frac{10}{24} = 0,4167. \tag{4}$$

## 4 Experiments and Results

In our experiments 18 volunteers participated in testing the system. Typing skills varied slightly among them - the majority of the group type on PC keyboard every day. Every volunteer had assigned unique login-id and password. The full name of particular individual was used as her/his login-id, since it is one of the most frequently typed phrase for most people.

In the first stage every participant performed 10 attempts of login-password authentication that will be evaluated by the system in order to calculate the model vector of digraphs and trigraphs as well as to collect the typing paths. Later on user performed another 20 logon attempts as valid user and 10 attempts as impostor (**trying to log on somebody's else account knowing login and password**).

In our experiments we had set the systems for different thresholds: 0.25, 0.3, 0.35 and 0.4. We focused on standard biometrics tests and therefore we calculated False Rejection Rate (*FRR*) and False Acceptance Rate (*FAR*) for each of the users.

We tested the system based on all the presented methods combined together. Results of our tests are presented in sections 4.1 and 4.2 followed by discussion of the results.

#### 4.1 FRR Tests and Results

Each user performed 20 logon attempts as valid user. The combined values of *FRR* varied from 0% to 55% and are presented in Table 1.

Unfortunately, usually after several successful attempts most of the users wanted to find out how the system behaves in case of sudden change of typing patterns and they 'test' the system trying to type in extremely different way than they used to. This behavior of users is inevitable in real-life applications and it definitely affected the *FRR* performance of the system.

**Table 1.** FRR results for all the combined methods

user	Combined FRR
user1	7.6923
user2	2.5000
user3	0.0000
user4	41.3043
user5	55.5556
user6	6.2500
user7	41.6667
user8	32.0000
user9	0.0000
user10	15.0000
user11	22.7273
user12	33.3333
user13	36.8421
user14	43.7500
user15	36.3636
user16	9.5238
user17	7.6923
user18	15.7895

## 4.2 FAR Tests and Results

In the second part of experiments a participant was asked to act as impostor. She/he was trying to logon on somebody else account [16]. In order to increase the number of logon attacks per single account, we randomly selected 10 out of 18 existing accounts to be attacked.

This decision was motivated by the fact that the number of participants (and thus samples) was limited (users were not willing to spend hours trying to hack somebody's else account). Bigger number of attacks per single account will picture more clearly the FAR, so smaller number of accounts to hack was the only reasonable solution.

The values of *FAR* were equal to %0 for all but 2 users (Table 2). For the 2 users it was possible for the impostor to logon with their password and biometrics characteristics with the probability 1.9% and 8.2%, respectively.

**Table 2.** FAR results for all the combined methods

user	Combined FAR
user1	<b>1.9230</b>
user2	<b>0.0000</b>
user6	<b>0.0000</b>
user8	<b>0.0000</b>
user9	<b>0.0000</b>
user10	<b>0.0000</b>
user14	<b>8.1649</b>
user15	<b>0.0000</b>
user17	<b>0.0000</b>
user18	<b>0.0000</b>

## 4.3 Discussion

In a password hardening application of keystroke dynamics (e.g. for e-banking) *FAR* is more important than *FRR* and therefore we think our results are satisfactory. Nevertheless some minor changes to our implementation could decrease *FRR*, which would make the system more user-friendly.

It is hard to determine which of the developed and implemented method gives the best performance for all users. The best solution is to make the logon algorithm adaptive. The algorithm should check which method gives the best performance for given user in order to give it the biggest weight while taking the access/no access decision. In case of non-adaptive implementation the best results were observed for thresholds: 0,25 for trigraphs and 0,3 for digraphs.

The threshold for digraphs and trigraphs should not be equal. It should be higher for digraphs and lower for trigraphs. It is also noticeable that longer char sets (trigraphs) have more stable statistics for a legitimate user (the standard deviation of particular trigraph's durations is small, and thus the distance calculated from the degree of disorder is smaller), but on the other hand they are easier to forge.

## 5 Applications

Password-based-only authentication works well for many applications and is quite simple to implement. Nevertheless, additional use of keystroke analysis could be encouraged in many situations, some of which are presented below:

- Password Hardening - Password hardening using keyboard statistics can be described as login-password pair combined with the collected typing features during the logon process. A system that implements the password hardening not only performs the password verification but also checks the typing patterns [17]. The main advantage of this approach is the significant increase of the security.
- Identity Verification - keyboard statistics could be introduced into any verification system right after the user's login-password pair typing stabilizes.
- Strong Authentication - root password, safety-critical systems and resources.
- Forgotten Passwords - algorithm could be used in forgotten password recovery.

## 6 Conclusion

In the article we presented our keystroke feature (characteristics) extraction methods. We also presented experimental results and proved that biometrics based on keystroke dynamics is capable of performing human identification tasks, especially in web applications demanding high level of security.

Keystroke dynamics are sensitive to the emotional and physical state of the person who is verified. Very poor typing skills are another factor which can affect the process of authentication. The good thing is that this method is very likely to achieve a high level of acceptance among ordinary users. Moreover, unlike other biometric systems which usually require additional hardware and thus are expensive to implement, biometrics based on keystroke dynamics is almost for free - the only hardware required is the keyboard.

## References

1. F. Monrose, A. Rubin, "Keystroke Dynamics as a Biometric for Authentication", *Future Generation Computer Systems*, vol. 16 , no. 4, 351 - 359, 2000.
2. M.S. Obaidat, B. Sadoun, "Keystroke Dynamics Based Authentication", in: *Biometrics: Personal Identification in Networked Society (Eds: A.K. Jain, R. Bolle, S. Pankanti, 1998.*
3. M.S. Obaidat, B. Sadoun, "Verification of Computer Users Using KEystroke Dynamics", *IEEE Trans. Syst., Man, Cybern.-Part B*, vol. 24, no. 2, 261-269, 1997.
4. G. Leggett, J. Williams, M. Usnick, "Dynamic Identity Verification via Keystroke Characteristics", *International Journal of Man-Machine Studies*, vol. 35 , no. 6, 859 - 870, 1991.
5. R. Gaines, W. Lisowski, S. Press, N. Shapiro, "Authentication by Keystroke Timing: some preliminary results", *Rand Report R-256-NSF*. Rand Corporation (1980)

6. F. Monrose, A. Rubin, "Authentication via Keystroke Dynamics", Conference on Computer and Communications Security , 48-56, 1997.
7. R. Joyce, G. Gupta, "User authorization based on keystroke latencies", Communications of ACM, vol. 33, no. 2, 168-176, 1990.
8. S. Bleha, C. Slivinsky, B. Hussein, "Computer-access security systems using keystroke dynamics", IEEE Trans. on Patt. Anal. Mach. Int, vol. 12, no. 12, 1217-1222, 1990.
9. M. Brown, S. J. Rogers, "User identification via keystroke characteristics of typed names using neural networks", International Journal of Man-Machine Studies, no. 39, 999-1014, 1993.
10. F. Bergadano, D. Gunetti, C. Picardi, "User Authentication through Keystroke Dynamics", ACM Transactions on Information and System Security, vol.5, no. 4, 367 - 397, 2002.
11. M. Brown, S. J. Rogers, "Method and apparatus for verification of a computer user's identification, based on keystroke characteristics", Patent Number 5,557,686, U.S. Patent and Trademark Office, Washington, D.C., Sept. (1996).
12. E. Yu, S. Cho, "Biometrics-based Password Identity Verification: Some Practical Issues and Solutions," XVth Triennial Congress of the International Ergonomics Association (IEA), Aug 24-29 2003, Seoul, Korea.
13. M. Tapiador, J.A. Sigenza, "Fuzzy Keystroke Biometrics On Web Security", Proc. of AutoID 1999.
14. <http://www.biopassword.com>
15. A.K. Jain, L. Hong, S. Pankanti, "Biometrics identification", Communications of the ACM, vol. 43, no. 2, 2000.
16. B. Schneier, "Inside risks: the uses and abuses of biometrics", Communications of the ACM, vol. 42, no. 8, 1999.
17. P. Mroczkowski, M. Choraś, "Keystroke Dynamics in Biometrics Client-Server Password Hardening System", Proc. of Advanced Computer Systems (ACS), vol. II, 75-82, Miedzyzdroje, Poland, October 2006.



# Protecting Secret Keys with Fuzzy Fingerprint Vault Based on a 3D Geometric Hash Table

Sungju Lee<sup>1</sup>, Daesung Moon<sup>2</sup>, Seunghwan Jung<sup>1</sup>, and Yongwha Chung<sup>1,\*</sup>

<sup>1</sup>Department of Computer and Information Science, Korea University, Korea  
{peacfeel, sksghksl, ychungy}@korea.ac.kr

<sup>2</sup>Biometrics Technology Research Team, ETRI, Daejeon, Korea  
daesung@etri.re.kr

**Abstract.** Biometrics-based user authentication has several advantages over traditional password-based systems for standalone authentication applications such as home networks. This is also true for new authentication architectures known as *crypto-biometric* systems, where cryptography and biometrics are merged to achieve high security and user convenience at the same time. Recently, a cryptographic construct, called *fuzzy vault*, has been proposed for crypto-biometric systems. In this paper, we propose an approach to provide both the automatic alignment of fingerprint data and higher security by using a 3D geometric hash table. Based on the experimental results, we confirm that the proposed approach of using the 3D geometric hash table with the idea of the *fuzzy vault* can perform the fingerprint verification securely even with one thousand chaff data included.

## 1 Introduction

Current cryptographic algorithms have a very high proven security, but they suffer from the key management problem: all these algorithms fully depend on the assumption that the keys will be kept in absolute secrecy. For example, in a home network environment, the secret keys for the family members may not be maintained securely. The most popular authentication mechanism used for key release is based on passwords, which are again cryptographic key-like strings but simple enough for users to remember. Simple passwords compromise security, but complex passwords are difficult to remember and expensive to maintain. Further, passwords are unable to provide non-repudiation: a subject may deny releasing the key using password authentication, claiming that her password was stolen[1].

Many of these limitations of password-based key release can be eliminated by incorporating biometric data. It is inherently more reliable than password-based key release as biometric characteristics cannot be lost or forgotten. Further, biometric characteristics are difficult to copy, share, and distribute, and require the person being authenticated to be present at the time and point of authentication. Thus, biometrics-based solution is a potential candidate to replace password-based solution, either for providing complete authentication mechanism or for securing the traditional cryptographic keys. In this paper, the fingerprint has been chosen as the biometrics for

---

\* Corresponding Author.

user authentication. It is more mature in terms of the algorithm availability and feasibility[2].

A cryptographic system and a biometric system can be merged in one of the following two modes[2-11]: (i) In a “loosely-coupled” mode of cryptography and biometrics, the biometric matching is decoupled from the cryptographic part. Biometric matching operates on the traditional biometric templates. (ii) In a “tightly-coupled” mode of cryptography and biometrics[6-11], biometrics and cryptography are merged together at a much deeper level. Biometric matching can effectively take place within cryptographic domain, hence there is no separate matching operation that can be attacked; positive biometric matching “extracts” the secret key from the conglomerate(key/biometric template) data. An example of this mode, called *fuzzy vault*, was proposed by Juels and Sudan[8].

In this paper, we focus on the *fuzzy vault*, especially *fuzzy fingerprint vault*. Recently, some implementations results for the fuzzy fingerprint vault have been reported by assuming that fingerprint features were pre-aligned. However, an automatic approach to align fingerprint features in the fuzzy vault domain needs to be developed, and it is challenging because the alignment should be performed in a non-invertible transformed domain. That is, an alignment should be performed between the enrollment template added by a lot of the “chaff” points and the input template without such chaff points. In [11], we proposed an approach of automatic fingerprint alignment by using the geometric hashing technique[12] which has been used for model-based object recognition applications. In this paper, we improve the secrecy of the proposed approach with a 3D geometric hash table. Based on the experimental results, we confirm that the proposed approach with a 3D geometric hash table can insert one thousand chaff points and protect the secret key more securely, and cryptography and biometrics can be merged to achieve higher security and user convenience at the same time.

## 2 Background

### 2.1 Key Protection with Fingerprint

Recently, Juels and Sudan[8] proposed the *fuzzy vault*, a new approach with applications similar to Juels and Wattenberg's fuzzy commitment[7], but is more compatible with partial and reordered data. In the fuzzy commitment, Alice can place a secret value  $k$  (e.g., private encryption key) in a vault and lock(secure) it using an unordered set  $A$ . Bob, using an unordered set  $B$ , can unlock the vault(access  $k$ ) only if  $B$  overlaps with  $A$  to a great extent. Note that, since this fuzzy vault can work with unordered sets(common in biometric templates, including fingerprint minutiae data), it is a promising candidate for crypto-biometric systems.

Based on the fuzzy vault, some implementations results for fingerprint have been reported. For example, Clancy, et al.[9] and Uludag, et al.[10] proposed a *fuzzy fingerprint vault*. Note that, their approaches inherently assume that fingerprints(the one that locks the vault and the one that tries to unlock it) are pre-aligned. This is not a realistic assumption for fingerprint-based authentication schemes, and limits the applicability of their approaches.

To solve the alignment problem, we have proposed an approach based on the geometric hashing technique[12]. Although our previous approach can solve the automatic alignment problem, the approach still has the problem of limited security. That is, the maximum number of chaff points for hiding the real fingerprint minutiae is limited by the size of the fingerprint sensor. For example, all the previous approaches[9-11] assumed the number of chaff points was 200. With the advance of the computing power provided by modern processors, the fuzzy fingerprint vault systems with this number of chaff points may not be secure. For higher security, we need an approach which allows more number of chaff points.

## 2.2 Geometric Hashing

In a model-based recognition system, a set of objects is given and the task is to find instances of these objects in a given scene. The objects are represented as sets of geometric features, such as points or lines, and their geometric relations are encoded using a minimal set of such features. The task becomes more complex if the objects overlap in the scene and/or other occluded unfamiliar objects exist in the scene.

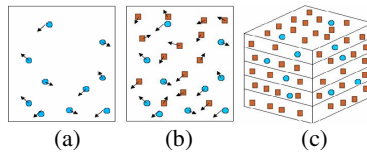
Geometric Hashing[12] offers a different and more parallelizable paradigm. It can be used to recognize flat objects under weak perspective. Because of such robustness, geometric hashing has been used for many applications. In the following, for the sake of completeness, we briefly outline the geometric hashing technique. Additional details can be found in [12].

## 3 Fuzzy Fingerprint Vault Based on a 3D Geometric Hash Table

In this section, we explain a proposed approach to generate protected fingerprint templates for higher security using a 3D hash table and perform fingerprint verification with the protected templates. To explain our approach, we first describe the fuzzy vault in more detail. We assume that Alice can place a secret value  $k$  in a vault and lock it using an unordered locking set  $L$ . Bob, using an unordered unlocking set  $U$ , can unlock the vault only if  $U$  overlaps with  $L$  to a great extent. The procedure for constructing the fuzzy vault is as follows: Secret value  $k$  is first encoded as the coefficients of some degree  $d$  polynomial in  $x$  over a finite field  $GF(q)$ . This polynomial  $f(x)$  is now the secret to protect. The locking set  $L$  is a set of  $t$  values  $l_i \in GF(q)$  making up the fuzzy encryption key, where  $t > d$ . The locked vault contains all the pairs  $(l_i; f(l_i))$  and some large number of chaff points  $(a_i, \beta_i)$ , where  $f(a_i) \neq \beta_i$ . After adding the chaff points, the total number of items in the vault is  $R$ . In order to crack this system, an attacker must be able to separate the chaff points from the legitimate points in the vault. The difficulty of this operation is a function of the number of chaff points, among other things. A legitimate user should be able to unlock the vault if they can narrow the search space. In general, to successfully interpolate the polynomial, they have an unlocking set  $U$  of  $d$  elements such that  $L \cap U$  contains at least  $d + 1$  elements. The details of the fuzzy vault can be found in [7, 8, 10, 11].

### 3.1 Fingerprint Templates Based on 3D Hash Table

For protecting template minutiae from the attacker, previous approaches[10, 11] using the fuzzy vault scheme store a number of chaff minutiae generated randomly as well as the real minutiae. However, the previous approaches[10, 11] have employed the number of the chaff minutiae ranged from 100 to 200, which means the difficulties of possible attacks. Furthermore, the max number of chaff minutiae employed in the previous approaches is determined by both the size of the fingerprint images captured and the possible degradation of the verification accuracy caused by the added chaff minutiae. With the advance of computing power, we need to add more number of chaff minutiae for higher security(*i.e.*, to make the attackers to difficult to separate the chaff minutiae from the real minutiae in the vault). Someone may add more number of chaff minutiae to the 2D fingerprint image captured. However, this simple solution does not work, because the 2D fingerprint image added with large number of chaff minutiae is too crowd to separate the chaff minutiae from the real minutiae even for a legitimate user. Therefore, we need a different approach to add more number of chaff minutiae in the template for higher security. In this paper, we propose an approach to use a 3D hash table, and Fig.1 shows examples of generated templates with typical fingerprint systems, fuzzy fingerprint vault, and the proposed approach.



**Fig. 1.** Example of Generated Templates; (a) Example of a Typical Template[2, 3]. (b) Example of a Protected Template based on Fingerprint Fuzzy Vault[10, 11](circles: real minutiae, rectangles: chaff minutiae). (c) Example of a Protected Template proposed with 3D Hash Table.

#### 3.1.1 Transform into the 3D and Selection of Chaff Minutiae

Fingerprint minutia represented by  $m_i = (x_i, y_i, \theta_i, t_i)$  is composed of three elements such as coordinates, angle, and type. First, we need to construct a 3D table for adding more number of the chaff minutiae. Because the fingerprint minutiae  $m_i$  consists of two coordinates( $x_i, y_i$ ), angle( $\theta_i$ ) where  $0 < \theta \leq 360$ , and type( $t_i$ ), the  $z$ -coordinate can be generated by angle, and fingerprint minutiae in the 3D table can be represented by  $m_i = (x_i, y_i, z_i, \theta_i, t_i)$  for a 3D table.

Note that, it is challenging to perform fingerprint verification with the protected template added by this number of chaff minutiae. To increase the verification accuracy of our fuzzy fingerprint vault system, we determine the positions and the angles of the added chaff minutiae carefully. For example, if the positions and the angles of the added chaff minutiae are similar to those of a real minutia of a legitimate user, then the corresponding input minutia of the legitimate user can be aligned with the added chaff minutiae resulting in an incorrect alignment. Therefore, we first define the acceptable margin  $\Delta d1$ ,  $\Delta d2$ , and  $\Delta d3$  for between  $x$ -/ $y$ -,  $y$ -/ $z$ -, and  $z$ -/ $x$ -coordinate in the 3D table, respectively. Since two minutiae within  $\Delta d1$ ,  $\Delta d2$ , and  $\Delta d3$  can be considered as a

matched minutiae pair, the positions of the added chaff minutiae need to be determined apart from the acceptable margin derived from any real minutiae for accurate fingerprint verification. Additionally, newly added chaff minutiae need to consider the positions of the already added chaff minutiae in order not to reveal them as chaff minutiae where  $z_{i-1} < \theta_i < z_{i+1}$ . However, the type information of the chaff minutiae is selected randomly, because it is less important than other information.

**3.1.2 Creation of Enrollment Minutiae Table**

Let  $m_i = (x_i, y_i, z_i, \theta_i, t_i)$  represent a minutia and  $L = \{m_i \mid 1 \leq i \leq r\}$  be a locking set including the real and chaff minutiae. In  $L$ , the real and chaff minutiae can be represented by  $G = \{m_i \mid 1 \leq i \leq n\}$  and  $C = \{m_i \mid n+1 \leq i \leq r\}$ , respectively. Note that, the 3D enrollment minutiae table is generated from  $L$ .

In the enrollment minutiae table generation stage, a 3D enrollment table is generated in such a way that no alignment is needed in the verification process for unlocking vault by using the geometric hashing technique. That is, alignment is pre-performed in the enrollment table generation stage. In verification process, direct comparisons without alignment are performed in 1:1 matching between the 3D enrollment minutiae table and an input fingerprint in order to select the real minutiae( $G$ ) only. Each step in the enrollment minutiae table generation stage is explained in detail in the following.

1) Reference Point Selection Step

In reference point selection step, a minutia is selected as the first minutia from the set of enrollment minutiae( $L$ ). The first minutia is denoted by  $m_1$  and the other remaining minutiae are denoted as  $m_2, m_3, \dots, m_n$ . At this moment, the minutia,  $m_1$ , is called *basis*.

2) Minutiae Transform Step

In minutiae transform step, minutiae  $m_2, m_3, \dots, m_n$  are aligned with respect to the first minutia  $m_1$  and quantized. Let  $m_i(1)$  denote the transformed minutiae, *i.e.*, the result of the transform of the  $j$ th minutia with respect to  $m_1$ . Also, let  $T_j$  be the set of transformed minutiae  $m_i(1)$ , *i.e.*,  $T_j = \{m_i(1) = x_i(1), y_i(1), z_i(1), \theta_i(1), t_i(1) \mid 1 < j \leq r\}$ , and  $T_j$  is called the  $m_j$ -transformed minutiae set. Equation (1) represents the translation and rotation such that features( $x_j, y_j, z_j, \theta_j, t_j$ ) of  $m_j$  is translated and rotated into  $(1, 1, 1, 1, t_j)$ . Let  ${}_{TR}m_i(1)$  denote the minutia translated and rotated from the  $j$ th minutia with respect to  $m_1$ . In addition, to reduce the memory space of the 3D hash table, the  $z_i$  coordinates are made from  $\theta_i / \alpha$ . In this paper,  $\alpha$  is set to 18, which causes  $0 \leq z_i \leq 20$ .

3) Repeat Step

Step 1) and step 2) are repeated for all the remaining minutiae. When step 1) and step 2) are completed for all the minutiae of the enrollment user, the 3D enrollment minutiae table is generated completely.

$${}_{TR}m_j(1) = \begin{pmatrix} {}_{TR}x_j(1) \\ {}_{TR}y_j(1) \\ {}_{TR}z_j(1) \\ {}_{TR}\theta_j(1) \\ {}_{TR}t_j(1) \end{pmatrix} = \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & 0 & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 & 0 \\ 0 & 0 & 1/\alpha & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_j - x_1 \\ y_j - y_1 \\ \theta_j \\ \theta_j \\ t_j \end{pmatrix}, \text{ where } 1 < j \leq r \tag{Eq. 1}$$

Finally, the locked vault contains all the real minutiae( $x_i, y_i, z_i, \theta_i, t_i, f(x_i, y_i)$ ) and some large number of the chaff minutiae( $x_j, y_j, z_j, \theta_j, t_j, f(x_j, y_j)$ ), where  $f(x_j, y_j) \neq \beta_j$ .

### 3.2 Fingerprint Verification with Protected Templates

After the enrollment process, the verification process to separate the chaff minutiae( $C$ ) from the real minutiae( $G$ ) in the 3D enrollment minutiae table should be performed. In the verification process, minutiae information(unlocking set  $U$ ) of a verification user is obtained and another 3D table, called *verification table*, is generated according to the geometric characteristic of the minutiae. Then, the 3D verification table is compared with the 3D enrollment minutiae table, and the subset of real minutiae is finally selected. Note that, the verification table generation stage is performed in the same way as in the enrollment process.

In comparing the enrollment and verification minutiae tables, the transformed minutiae pairs with the same coordinates, the same angle, and the same type are determined. The minutiae pairs having the maximum number and the same basis are selected as the subset of real minutiae( $G$ ). Also, any additional alignment process is not needed because pre-alignment with each minutia is executed in the enrollment and verification minutiae table generation stage.

## 4 Implementation Details and Experimental Results

For the purpose of evaluating of the accuracy of the proposed approach, a data set of 1,600 fingerprint images composed of four fingerprint images per one finger was collected from 400 individuals by using the optical fingerprint sensor[13]. The resolution of the sensor was 500dpi, and the size of captured fingerprint images was 248×292.

To evaluate the effect of the fuzzy fingerprint vault using the 3D hash table, we measured the performance for 100, 200, and 400 chaff minutiae with the 2D hash table, generated straightforwardly from the 2D fingerprint images, in addition to the performance for 100, 400, and 1000 chaff minutiae with the 3D hash table. Due to the acceptable margin explained in Section 3, 400 is the maximum number of chaff minutiae which can be added by using the 2D hash table.

In this paper, we fixed the size of the enrollment minutiae table which was  $255 \times 255 \times 360/\alpha$  by using scaling operators. Then, we determined the acceptable margin of coordinates(4), distance of the minutiae(4), acceptable margin of angle(5), and  $\alpha$ (18). Also, the enrollment minutiae table had 1000 chaff(max) and 36 real(average) points. Thus, the size of the enrollment minutiae table was  $255 \times 255 \times 360/18B + 1036 \times 1035B = 2.24MB$ . Also, the average enrollment time was 13 second and the average verification time was 0.9 second. Note that, the enrollment procedure is executed off-line and only once, the online, repeated verification procedure can be performed in the real time.

To compare of the verification accuracy of the proposed approach using the 3D hash table with that of the 2D hash table, we measured the verification performance(FRR and FAR) with the various parameters such as degree of the polynomials and the number of the chaff minutiae. Also, in the genuine experiment, a

matched chaff minutiae was removed by using error-correcting code such as a Reed-Solomon decoder. Table 1 shows the experimental result for the verification performance. We can consider that the input minutiae are from a genuine if the unlocking set reconstructs the correct polynomial. If we use a 10-degree polynomial with 11 coefficients, 11 unique projections are required for unlocking the 10-degree polynomial. As shown in Table 1, 4,364 of the 4,800 attempts were able to successfully unlock the vault in the 10-degree and 1000chaff minutiae.

**Table 1.** Experimental Results for the Various Parameters

Degree of Polynomial	Based on 2D Hash Table						Based on 3D Hash Table					
	100 chaff minutiae		200 chaff minutiae		400 chaff minutiae		100 chaff minutiae		400 chaff minutiae		1000 chaff minutiae	
	FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR
8	0.065	0.006	0.070	0.009	0.101	0.011	0.011	0.025	0.024	0.018	0.066	0.005
9	0.098	0.003	0.104	0.004	0.131	0.006	0.019	0.015	0.033	0.010	0.075	0.003
10	0.144	0.002	0.151	0.002	0.174	0.004	0.031	0.010	0.048	0.006	0.091	0.001

We can also analyze the security of the system mathematically as in [10]. Assume that we have an attacker who does not use real fingerprint data to unlock the vault; instead he tries to separate real minutiae from chaff minutiae in the vault using brute-force. If we use a 10-degree polynomial with 11 coefficients, attacker needs at least 11 minutiae to reconstruct the correct polynomial. The vault has 1,036 minutiae(36 of them are real, remaining 1,000 are chaff); hence there are a total of  $C(1036,11) \approx 3.504 \times 10^{25}$  combinations with 13 elements. Only  $C(36,11) \approx 6.008 \times 10^8$  of these combinations will reveal the secret(*i.e.*, unlock the vault). Thus, it will need an average of  $5.8335 \times 10^{16} (=C(1036,11)/C(36,11))$  evaluations for an attacker to crack the vault. On the contrary, if the vault has 436 minutiae(36 of them are real, remaining 400 are chaff), an average of  $3.974 \times 10^{12} (=C(436,11)/C(36,11))$  evaluations for an attacker to crack the vault is needed.

## 5 Conclusions

In this paper, an approach to align fingerprint features automatically in the domain of the fuzzy fingerprint vault has been proposed. By employing the geometric hashing technique which has been used for model-based object recognitions, we can achieve automatic alignment of fingerprint features in the domain of the fuzzy vault. To evaluate the effectiveness of our approach, we conducted preliminary experiments. Based on the experimental results, our approach by using the 3D hash table generated for the fuzzy fingerprint vault can align the fingerprint features accurately in real-time and can be integrated with any fuzzy fingerprint vault systems. Currently, we are conducting more experiments to obtain optimal design parameters and to reduce the size of the hash table.

## Acknowledgement

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the Chung-Ang University HNRC-ITRC(Home Network Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

## References

1. W. Stallings, *Cryptography and Network Security*, Pearson Ed. Inc., 2003.
2. D. Maltoni, et al., *Handbook of Fingerprint Recognition*, Springer, 2003.
3. R. Bolle, J. Connell, and N. Ratha, "Biometric Perils and Patches," *Pattern Recognition*, Vol. 35, pp. 2727-2738, 2002.
4. S. Prabhakar, S. Pankanti, and A. Jain, "Biometric Recognition: Security and Privacy Concerns," *IEEE Security and Privacy*, pp. 33-42, 2003.
5. U. Uludag, et al., "Biometric Cryptosystems: Issues and Challenges," *Proc. of IEEE*, Vol. 92, No. 6, pp. 948-960, 2004.
6. C. Soutar, et al., "Biometric Encryption – Enrollment and Verification Procedures," *Proc. SPIE*, Vol. 3386, pp. 24-35, 1998.
7. A. Juels and M. Wattenberg, "A Fuzzy Commitment Scheme," *Proc. of ACM Conf. on Computer and Comm. Security*, pp. 28-36, 1999.
8. A. Juels and M. Sudan, "A Fuzzy Vault Scheme," *Proc. of Symp. on Information Theory*, pp. 408, 2002.
9. T. Clancy, N. Kiyavash, and D. Lin, "Secure Smartcard-based Fingerprint Authentication," *Proc. of ACM SIGMM Multim., Biom. Met. & App.*, pp. 45-52, 2003.
10. U. Uludag, S. Pankanti, and A. Jain, "Fuzzy Fingerprint Vault," *Proc. of Workshop Biometrics: Challenges arising from Theory to Practice*, pp. 13-16, 2004.
11. Y. Chung, et al., "Automatic Alignment of Fingerprint Features for Fuzzy Fingerprint Vault," *LNCS 3822*, pp. 358-369, 2005.
12. H. Wolfson and I. Rigoutsos, "Geometric Hashing: an Overview," *IEEE Computational Science and Engineering*, Vol. 4, pp. 10-21, Oct.-Dec. 1997.
13. D. Ahn, et al., "Specification of ETRI Fingerprint Database(in Korean)," *Technical Report – ETRI*, 2002.



# Face Recognition Based on Near-Infrared Light Using Mobile Phone

Song-yi Han<sup>1</sup>, Hyun-Ae Park<sup>2</sup>, Dal-ho Cho<sup>1</sup>, Kang Ryoung Park<sup>3</sup>,  
and Sangyoun Lee<sup>4</sup>

<sup>1</sup> Dept. of Computer Science, Sangmyung University,  
Biometrics Engineering Research Center

7 Hongji-Dong, Jongro-gu, Seoul, Republic of Korea

<sup>2</sup> Image Sensor & Vision Technology™ Pixelplus Co., Ltd.

<sup>3</sup> Division of Digital Media Technology, Sangmyung University,  
Biometrics Engineering Research Center

7 Hongji-Dong, Jongro-gu, Seoul, Republic of Korea

<sup>4</sup> Biometrics Engineering Research Center,

Dept. of Electrical and Electronic Engineering, Yonsei University,

134 Shinchon-dong, Seodaemon-ku, Seoul, Republic of Korea

ttlskylove@smu.ac.kr

**Abstract.** Recently, many companies have attempted to adopt biometric technology in their mobile phones. In this paper, we propose a new NIR (Near-Infra-Red) lighting face recognition method for mobile phones by using megapixel camera image. This paper presents four advantages and contributions over previous research. First, we propose a new eye detection method for face localization for mobile phones based on corneal specular reflections. To detect these SRs (Specular Reflections) (even for users with glasses), we propose successive On/Off activation of the dual NIR illuminators of mobile phone. Second, because the face image is captured by the NIR illuminator, the nose area can be highly saturated, which can degrade face recognition accuracy. To overcome this problem, we use a simple logarithmic image enhancement method, which is suitable for mobile phones with low processing power. Third, considering the low processing speed of mobile phones, we adopt integer-based PCA (Principal Component Analysis) method for face recognition excluding floating-point operation. Fourth, by comparing the recognition performance using the integer-based PCA to those using LDA (Linear Discriminant Analysis) and ICA (Independent Component Analysis) methods, we could know that the integer-based PCA showed better performance apt for mobile phone with NIR image.

## 1 Introduction

With recent developments of mobile phones, the security of personal information on mobile phones is becoming more important. Therefore, fingerprint recognition phones have been already manufactured. However, they are more expensive and bigger than conventional mobile phones because they require an additional fingerprint sensor as well as a DSP (Digital Signal Processor) chip for fingerprint recognition. In addition, because the fingerprint sensor should be small due to the size limitation of mobile

phone, it may lead to unreliable authentication performance. So, fingerprint recognition phones have not become widely popular yet. However, with rapid developments of mobile phone, many companies have adopted a built-in mega-pixel camera in mobile phone and it can give the possibility of face and iris recognition on mobile phone without additional sensor. Our final goal of research is to develop a multimodal biometric system based on face and iris recognition on a mobile phone. For that, we propose a new NIR (Near-Infra-Red) lighting face recognition method apt for mobile phones in this paper. When using a mega-pixel camera, the iris region contains enough pixel information to be identified in the captured face image. This makes it possible to identify both face and iris with mega-pixel face image. The reasons for adopting multi-modal biometric systems in mobile phones are like these. Although iris recognition shows higher levels of accuracy, the accuracy decreases when authenticating users with small eyes. In addition, in outdoor locations with lots of sunlight (which is often the case when using mobile phones), many ghost spots can appear in iris region, which can also degrade recognition performance. In the mean while, the accuracy of conventional face recognition is much lower compared to iris recognition. So, multi-modal biometrics can solve the above problems.

Most previous researches about face detection and recognition have been done under conditions of visible light. Due to the degraded performance caused by changes in environmental visible light, the researches of [1-2] used FIR (Far Infra-Red) lighting for face recognition, but it has a severe problem of requiring very expensive thermal camera. So, some researches have been done with NIR light [3-6]. Stan Z. Li *et al.* proposed a face detection and recognition method based on NIR lighting [3]. However, this was a learning-based face/eye detection method, which took too much processing time when adopting it to mobile phones. In addition, when a large SR occurred in the eye region of users with glasses, the trained eye detector could not locate the eye region. Also, that used the learning-based face recognition method and the floating point Chi-square distance, which require too much processing power to be used in mobile phones [3]. The researches of [4-5] used an active NIR method to overcome the effect of illumination variations in face recognition. However, single illuminator was used to detect eye feature positions based on a bright pupil image. This may lead to many imposter SRs (Specular Reflection, whose gray level is similar to bright pupil) on the scratched glasses surface in case of users with glasses. Consequently, it was difficult to detect eye positions for face localization [4-5]. In addition, LDA (Linear Discriminant Analysis) and SVM (Support Vector Machine) methods were used for face recognition. These methods were based on a floating-point operation, which took too much processing time in mobile phones [5]. The research described in [6] also performed face detection on NIR face images, but because a single illuminator was used to detect eye features, it could not solve the problem of imposter SRs in case of user with glasses. Diego A. Socolinsky *et al.* proposed the fusion method of thermal infrared and near infrared face images for face recognition, which also required too expensive device for capturing thermal image and too complicated processing algorithm to be applied to mobile phone [7]. Another approach to adopt face detection technology on mobile phone was introduced by Viola and Jones [2]. They used AdaBoost face classifier and tested it on mobile phones such as Nokia 7650 (CPU of 104 MHz) and Sony-Ericsson P900 (CPU of 156 MHz), using an input image of 344×288 pixels. However, they used face image in

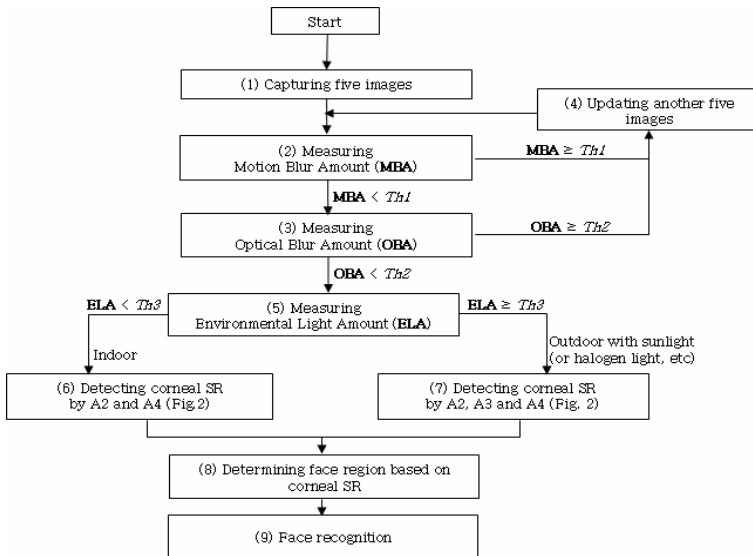
visible lighting (which was affected by environmental lighting condition) and an additional DSP chip for mobile phone, which led to an increase in total cost. K.H.Pun *et al.* proposed the extensive profiling method to enhance the processing speed of face authentication in mobile phone, but they used fixed point processing, which still required somewhat long processing time of 5 seconds for face authentication [8].

To overcome such problems of previous researches, we propose a new NIR lighting face detection and recognition method apt for mobile phones.

## 2 Proposed Face Detection and Recognition Method

### 2.1 Face Localization Based on the Corneal SR (Specular Reflection)

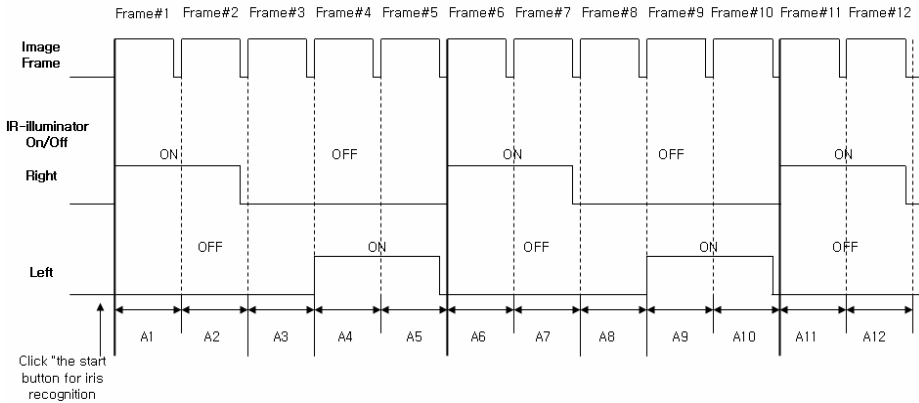
We first detect the face region for size normalization. In some kinds of mobile phone, user can see his face on the display window when he captures his face by mobile phone camera. However, in other kinds such as Samsung SCH-V770 [9], the display window is attached on the opposite side of camera (for both face and iris recognition, a mega-pixel camera should be used and in most cases, the mega-pixel camera is attached on the opposite side of display window without the rotating mechanism due to the complex camera structure) and user cannot see his face when capturing his face. In such a case, the user's face and eye region should be detected automatically.



**Fig. 1.** Flowchart of the proposed method

To locate the face region robustly, we propose the method of detecting the corneal SR with dual NIR illuminators [10]. An overview of the proposed method is shown in Fig.1. In this process, the user first initiates face recognition by clicking the “start” button of the mobile phone. Then, the camera micro-controller alternatively turns on

and off the dual (left and right) IR illuminators. In this case, the dual illuminators repeatedly turn on and off, synchronized with every frame of camera image. As shown in Fig.2, when only the right IR illuminator turns on, two facial images (Frame #1, #2) are captured. Another image (Frame #3) is captured when both illuminators turn off. After that, two additional facial images (Frame #4, #5) are captured again when only the left IR illuminator turns on. So, it is possible to obtain five successive images, as shown in Fig. 1(1) and Fig. 2. This illumination On/Off scheme of capturing five facial images is iterated successively as shown in Fig. 2. When Frames #1 ~ #5 do not meet our predetermined threshold for motion and optical blurring (as shown in Fig. 1(2), (3)), another five images (Frame #6 ~ #10) are captured (Fig. 1 (4)). The size of the captured original image is 3072×2304 pixels. To reduce the processing time, the captured eye region images are 1/6 down-sampled and the amount of motion blurring in the input image is checked, as shown in Fig. 1(2). The motion blur amount (**MBA**) can be calculated by the difference image between the two illuminator-On images such as Frame #1 and #2 (or Frame #4 and #5). If the calculated **MBA** is greater than the predetermined threshold (*Th1* as shown in Fig. 1) (we obtained 4 as the threshold by experiment), we can determine that the input image is too blurred to be recognized. After that, our system checks the amount of optical blurring (**OBA**) by checking the focus value of the A2 and A4 images in Fig. 2, as shown in Fig.1(3).



**Fig. 2.** The alternatively On/Off scheme of the dual IR-illuminators

We use the focus checking method as proposed by Kang *et al.* [11]. The calculated focus value is compared to the predetermined threshold. If either focus value of A2 or A4 is below the threshold (*Th2* as shown in Fig. 1) (we obtained 70 as the threshold by experiment), we regard the input image as defocused and then use another five captured images, as shown in Fig.1(4). Our system then calculate the Environmental Light Amount (**ELA**) of the illuminator-Off image (the average gray level of A3 as shown in Fig. 2) to check whether any external sunlight exists or not in the input image (as shown in Fig. 1 (5)). Because we attach an IR-Pass filter to the front of the

camera lens, the image brightness is not affected by any visible light. So, in indoor environments, the average gray level of the illuminator-Off image (A3) is very low (our experiments showed that it was below 50 (*Th3*)). However, sunlight includes a large amount of IR light and in outdoor environments with lots of sunlight, the average gray level of the illuminator-Off image (A3) increases to more than 50 (*Th3*). After that, our systems detect the corneal specular reflections in the input image. In indoor environments ( $ELA < Th3$  as shown in Fig.1 (6)), corneal SR detection is performed using the difference image between A2 and A4 in Fig. 1(6). In general, many imposter specular reflections (with similar gray levels to genuine specular reflections on the cornea) may occur in case of users with glasses. This makes it difficult to detect genuine specular reflections on the cornea (inside the pupil region). Our object is to distinguish between the specular reflections on cornea and glasses. So, we use a difference image in order to detect corneal specular reflections easily. The reason for this is because the genuine corneal specular reflections show the characteristics of existing as pair horizontally in the difference image, and their inter-distance in the image is much smaller than that of other imposter specular reflections on the surface of the glasses (due to the curvature radius of the cornea being much smaller than that of glasses). Also, we can suppose that corneal specular reflections have characteristics that they occur on pupil area in case that Z distance between eye and illuminator is far and they occur on iris region when the Z distance becomes short. That is due to the geometry of camera, illuminator and eye. However, in outdoor environments with lots of sunlight, SR is detected using the difference image between A2 and A4 (in this case, the gray level of each pixel in A2 and A4 is subtracted by that of same pixel in A3 in order to reduce the effect of outdoor sunlight from A2 and A4 because A3 is only illuminated by outdoor sunlight) as shown in Fig. 1(7) and Fig. 2. From the difference image, we are able to obtain the accurate center position of the genuine SRs based on the edge image by using the 3×3 Prewitt operator, component labeling and circular edge detection. Based on the detected position of the genuine SR in the eye cornea, the face region is determined based on the inter-distance between the two eyes and the geometric ratio between the eyes and the mouth. Then, the face region is down-sampled as 30×30 pixels. And light normalization as well as face recognition is performed.

## 2.2 Lighting Normalization

Infrared light images often display a bright, sometimes seemingly over-exposed, region around the image center [12]. This means that the bright center of an image has a problem in that a little information is lost by saturation. To solve this problem, homomorphic filtering was proposed by Wen-Hung Liao *et al.* [12], but this method requires complex FFT (Fast Fourier Transform) processing (based on the floating-point operation) which is not apt for mobile phones with low processing power. To overcome this problem and normalize over-exposed lighting, we propose a brightness normalization method based on logarithmic equation for real-time processing on mobile devices. To enhance processing speed in mobile phone, we use a look-up table of logarithmic equation, which is obtained in advance. To compare the processing complexity, we define the complexity ( $O(\ )$ ) as the multiplication number. The complexity of Wen-Hung Liao *et al.*'s method based on FFT is  $O(n \log_2 n)$ . Whereas,

because of using lookup table, the complexity of our algorithm based on logarithm is  $O(0)$ . In actual case, by using  $30 \times 30$  images, the complexity of FFT and logarithm are  $O(147.207)$  and  $O(0)$ , respectively. By using the light normalization method based on logarithmic equation, the gray level of the dark side area of the input image increases and its contrast becomes great, whereas the increasing amount of gray level of the highly saturated center region does not become great.

### 2.3 Face Recognition by Using the Integer-Based PCA Method

Considering the low processing speed of mobile phones, we propose an integer-based PCA (Principal Component Analysis) method for face recognition excluding the floating-point operation. PCA considers image elements as random variables with Gaussian distribution and minimizes second-order statistics [13]. Clearly, for any non-Gaussian distribution, the largest variances would not correspond to the PCA basis vectors.

In the meanwhile, ICA (Independent Component Analysis) [14] minimizes both second-order and higher-order dependencies in the input data and attempts to find the basis along which the data are statistically independent [13]. ICA uses multiple-trained bases with higher statistics [14]. Because higher statistics requires complex calculations, ICA takes too much processing time on mobile devices.

LDA (Linear Discriminant Analysis) [15] finds the vectors in the underlying space that best discriminate between the classes. LDA is reported to have better accuracy than PCA when more trained images are used [16]. However, for mobile phones with NIR illuminators, we cannot obtain many images for training, whereas we can collect a lot of images from many open face databases in visible lighting. Consequently, PCA can show better accuracy than LDA (see the results in Sect. 4). Based on those reasons, we use PCA for face recognition in this research.

Of course, PCA is reported to be sensitive to lighting conditions, but in our case, because we use an IR pass filter in front of the camera lens (for iris recognition) and NIR lighting for uniform illumination, the lighting changes are not great compared to those of face recognition by visible lighting. However, conventional PCA requires a floating-point operation, which takes too much processing time on mobile devices. To solve this problem, we propose an integer-based PCA method. After obtaining the floating-point eigenfaces by PCA training, we convert them into integer eigenfaces by multiplying them by  $10^6$  (actually, we use the bit shifting method of 20 bits for more fast speed). The multiplier ( $10^6$ ) was determined to obtain the minimum EER (Equal Error Rate) of face recognition. Here, the EER means the error rate when the FAR (False Acceptance Rate) is the same as the FRR (False Rejection Rate) with minimum error. The FAR represents the error rate of accepting the un-enrolled user as the enrolled one. The FRR does the error rate of rejecting the enrolled user as the un-enrolled one. The number of eigenfaces was determined to have a minimum EER of face recognition as 100 eigenfaces. In conventional PCA methods, because the input image should be also normalized by mean and standard deviations, the gray level of the input image has a range of 0 to 1. This is also converted to an integer value by bit shifting of 7 bits. Then, because both the eigenfaces and input face images have integer values, the calculated eigen-coefficients also have integer values. To consider the low processing power of mobile phones, we use the Euclidian distance for

matching. The threshold for accepting or rejecting user was determined (based on the Bayesian rule) to have a minimum EER of face recognition.

## 4 Experiment Results

To test proposed algorithm, we captured face images with 7 mega-pixel (3072×2304 pixels) camera of Samsung SCH-V770 mobile phone [9] with dual NIR illuminators (wavelength of 830nm). The distance between the camera and face was about 30 ~ 40cm. We obtained 350 images from 50 classes. Each class contained seven facial images: four images for neutral expression, one image for smile, one image for frown and one image for surprise. Half the images were used for PCA training and the others were used for recognition testing.

### 4.1 The Accuracy of Eye Detection Algorithm

We measured the accuracy of our eye detection algorithm. Tests were performed with the above 350 face images. These images were divided into the following six categories: in case of indoor lighting conditions (223 lux.), images without eyeglasses and contact lens (85 images), those with eyeglasses (70 images), and those with contact lens (20 images). And in outdoor lighting conditions with lots of sunlight (1,394 lux.), images without eyeglasses and contact lens (85 images), those with eyeglasses (70 images), those with contact lens (20 images). Experimental results showed that the successful eye detection rate was 99% (about images without eyeglasses in indoor and outdoor conditions) and 98.8% (about images with eyeglasses in indoor and outdoor conditions). From them, we could know that the detection rate was not degraded irrespective of user's wearing glasses due to our mechanism of illuminate-On & Off activation (see Sect. 2.1). Also, the detection rate in outdoor lighting conditions was almost the same as that in indoor.

### 4.2 The Performance of Proposed Brightness Normalization Method

We tested EER with or without our brightness normalization method based on logarithmic equation (see Sect. 2.2). With the normalization procedure, the EER was 14.79 % and without it, the EER was 16.43 %. Therefore, we discovered that the proposed normalization method can enhance recognition accuracy.

### 4.3 The Face Recognition Accuracy Using NIR Images

We compared the recognition performance of using NIR images to that using conventional visible face images. For visible face images, we used MPEG database [17]. The MPEG database consists of 300 images from 100 classes. Each class contains one neutral, one smile and one frown image excluding the images having illumination change. Experimental results showed that the EER with MPEG database was 14.81 % and that with NIR images was 14.8 %, respectively. In this case, NIR images also included one neutral, one smile and one frown images (which is the same condition as MPEG database). From that, we can know the accuracy with NIR images was almost same to that with MPEG database. Especially, the EER with NIR images

in case of including only neutral image was 12.65 % (because MPEG database includes only one neutral image, the comparative EER could not be measured) and that in case of including one neutral, one smile, one frown and one surprise NIR images was 16.11 % (because MPEG database does not include frown image, the comparative EER could not be measured).

#### 4.4 The Recognition Accuracy Using Floating-Point PCA

We compared the recognition accuracy of floating-point PCA to integer-based method. The EER of floating point and integer point PCA were 14.79 % (in case of only including neutral image, it was 12.65 %) and 14.81 % (in case of only including neutral images, it was 12.66 %), respectively. From that, we can see that the accuracy of integer-based PCA is almost the same as that using floating-point method.

#### 4.5 The Recognition Performance Using Integer-Based PCA, LDA and ICA

We compared the recognition accuracy of integer-based PCA method to that of LDA and ICA. The EER of LDA, ICA and integer point PCA were 15.32 %, 14.78 % and 14.81 %, respectively. Those in case of only including neutral images are 13.19 %, 12.62 % and 12.66 %, respectively. From that, we could know the accuracy of proposed method with NIR face image was better than that of LDA. Though the accuracy of ICA was a little better than ours, it takes much processing time than ours (the processing time of ICA in desktop PC and PDA (Personal Digital Assistant) were 82.23 ms and 556 ms). In case of using integer-based ICA, the processing time was much reduced as 34 ms (in desktop PC) and 230 ms (in PDA), but the EER was much increased as 19.1 % (in case of only including neutral images, it was 17.08 %). That is because ICA uses local information for face recognition and the integer-based ICA kernel loses its fine and local decomposition capability much more than integer-based PCA which uses global information of face.

**Table 1.** Comparative processing time on Desktop PC and PDA (unit: ms)

	Desktop PC		PDA	
	Floating-Point	Integer-Point	Floating-Point	Integer-Point
Eye Detection & Face Localization	10	10	21.98	21.98
Face Recognition	5.55	1.92	233.68	57.57
Total time	15.55	<b>11.92</b>	255.66	<b>79.55</b>

To simulate the processing time on a mobile phone, we tested our algorithm on PDA (HP iPAQ hx2750 [18]) because ARM core of PDA is the same as that of a mobile phone and neither one includes floating point co-processor. As Table 1, we could see that the processing speed of integer-based method was more than three times faster than that using floating-point one in PDA and it was much faster than previous work [8]. Because eye detection & face localization did not include floating point operation, the processing time was same in either case.



## 5 Conclusion and Future Work

In this paper, we have proposed a new NIR lighting face recognition method apt for mobile phones. In future work, we plan to test our algorithm on mobile phones and combine face and iris recognition with more field tests.

**Acknowledgements.** This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center (BERC) at Yonsei University.

## References

1. Shi-Qian Wu, et al., : Infrared Face Recognition by Using Blood Perfusion Data, Lecture Notes in Computer Science, Vol. 3546, (2005) 320-328
2. [www.idiap.ch/pages/contentTxt/Demos/demo29/face\\_finderfake.html](http://www.idiap.ch/pages/contentTxt/Demos/demo29/face_finderfake.html) (accessed on 09.01)
3. Stan Z. Li, et al., : Highly Accurate and Fast Face Recognition Using Near Infrared Images, Lecture Notes in Computer Science, Vol. 3832, (2006) 151-158
4. C.H. Morimoto, et al., : Real-time Multiple Face Detection Using Active Illumination, Proceedings of ICAFG, (2000), 8-13
5. Xuan Zou, et al., : Ambient Illumination Variation Removal by Active Near-IR Imaging, Lecture Notes in Computer Science, Vol. 3832, (2006) 19-25
6. J. Dowdall, et al., : Face Detection in the Near-IR Spectrum, Image and Vision Computing, Vol. 21 (2003) 565-578
7. Diego A. Socolinsky et al., : Face Recognition in Low-light Environments Using Fusion of Thermal Infrared and Intensified Imagery, Proceedings of SPIE, Vol. 6206, 2006
8. K. H. Pun et al., : A Face Authentication System for Mobile Devices: Optimization Techniques, Proceedings of SPIE, Vol. 5684, pp. 265-273, 2005
9. [www.samsung.com](http://www.samsung.com) (accessed on 09.01)
10. Hyun-Ae Park, et al. : Robust Iris Locating Method for Iris Recognition in Mobile Phone based on Corneal Specular Reflection and AdaBoost Classifier, Pattern Recognition Letters, (2006) submitted
11. Byung Joon Kang, Kang Ryoung Park, : A Study on Fast Iris Restoration Based on Focus Checking, LNCS, Vol. 4069, (2006) 19-28
12. Wen-Hung Liao; Dai-Yun Li : Homomorphic Processing Techniques for Near-infrared Images. Proceedings of ICASSP, Vol.3 (2003) 461-464
13. M. Turk et al., : Eigenfaces for Recognition. Journal of Cognitive Neuroscience, Vol. 3, No. 1 (1991) 71-86
14. M.S. Barlett, J.R. Movellan, T.J. Sejnowski : Face Recognition by Independent Component Analysis. IEEE Trans. on Neural Networks, Vol. 13, No. 6 (2002) 1450-1464
15. P.Belhumeur, J. Hespanha, D.Kriegman : Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. on PAMI, Vol. 19, No. 7 (1997) 711-720
16. A.M.Martinez, A.C.Kak : PCA versus LDA. IEEE Trans. on Pattern Analysis and Machine Intelligence Vol. 23 , No. 2 (2001) 228-233
17. MPEG, Call for Proposals for Face Recognition Technology, ISO/IEC JTC1/SC29/WG11/ N3676, La Baule, October 23-27, 2000.
18. [www.hp.com](http://www.hp.com) (accessed on 09.01)

# NEU-FACES: A Neural Network-Based Face Image Analysis System

Ioanna-Ourania Stathopoulou and George A. Tsihrintzis

Department of Informatics  
University of Piraeus  
Piraeus 185 34, Greece  
{iostath, geoatsi}@unipi.gr

**Abstract.** Towards building more efficient human control interactive systems, we developed a neural network-based image processing system (called NEU-FACES), which first determines automatically whether or not there are any faces in given images and, if so, returns the location and extent of each face. Next, NEU-FACES uses neural network-based classifiers, which allow the classification of several facial expressions from features that we develop and describe. NEU-FACES is fully implemented and evaluated to assess its performance.

## 1 Introduction

In the design of advanced human control interactive systems, the variations of the emotions of human users during the interaction should be taken into consideration. In human-to-human interaction and interpersonal relations, facial expressions play a significant communicative role because they can reveal information about the affective state, cognitive activity, personality, intention and psychological state of a person and this information may in fact be quite difficult to mask. Indeed, facial expressions corresponding to the “neutral”, “smile”, “sad”, “surprise”, “angry”, “disgust” and “bored-sleepy” psychological states are the most common in humans and their identification and classification may help in revealing a person’s intention, truthfulness of his/her statements or ultimate goal of his/her actions.

The task of processing facial images generally consists of two steps: a *face detection* step, which determines whether or not there are any faces in an image and, if so, returns the location and extent of each face, and a *facial expression classification step*, which attempts to recognize the expression formed on a detected face. These problems are quite challenging because faces are non-rigid and have a high degree of variability in size, shape, color and texture. Furthermore, variations in pose, facial expression, image orientation and conditions add to the level of difficulty of the problem. The task of determining the true psychological state of the person using an image of his/her face is complicated further by the problem of *pretence*, i.e. the case of the person’s facial expression not corresponding to his/her true psychological state. The difficulties in facial expression classification by *humans* and some indicative classification error percentages are illustrated in [1].

Previous attempts to address similar problems of face detection and facial expression classification in images have followed two main directions in the literature: (1) *methods based on face features* [2-5] and (2) *image-based representations* of the face [6-8]. Our system follows the first direction (feature-based approach) and has been developed over the past two years [9-13]. Specifically, the face detection algorithm currently used was developed and presented in [9], while the facial expression classification algorithms are evolved and extended versions of those gradually developed and presented in [10-14].

In the present version, NEU-FACES covers a wider set of facial expressions, for which good classification features had to be found, while the underlying neural networks have been re-structured and re-trained. This set of expressions was determined based on observations made in the reactions of persons when they use the computer. From these observations, the facial expressions that were extracted were the “neutral”, “smile”, “sad”, “surprise”, “angry”, “disgust” and “bored-sleepy” facial expression.

More specifically, the paper is organized as follows: in Sect. 2, we analyze the facial expression classification module of NEU-FACES. In Sect. 3, we illustrate and evaluate the performance of NEU-FACES. We draw conclusions and point to future work in Sects. 4 and 5, respectively.

## **2 Facial Expression Classification Subsystem**



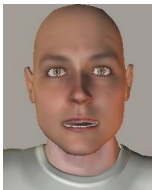


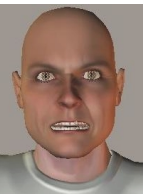
### **2.1 Discriminating Features for Facial Expressions**

For the classification task, we gathered and studied a dataset of 1400 images of facial expressions, which corresponded to 200 different persons forming the “neutral” and the six expressions: “smile”, “sad”, “surprise”, “angry”, “disgust” and “bored-sleepy”. We use the “neutral” expression as a template, which can somehow be deformed into other expressions. From our study of these images, we identified significant variations between the “neutral” and other expressions, which can be quantified into a classifying feature vector. Typical such variations are shown in Table 1.

### **2.2 The Feature Extraction Algorithm**

The feature extraction process in NEU-FACES converts pixel data into a higher-level representation of shape, motion, color, texture and spatial configuration of the face and its components. We extract such classification features on the basis of observations of facial changes that arise during formation of various facial expressions, as indicated in Table 1. Specifically, we locate and extract the corner points of specific regions of the face, such as the eyes, the mouth and the eyebrows, and compute variations in size or orientation from the “neutral” expression to another one. Also, we extract specific regions of the face, such as the forehead or the region between the eyebrows, so as to compute variations in texture. The extracted features are illustrated in Fig. 1.

**Table 1.** Formation of facial expressions via deformation of the neutral expression

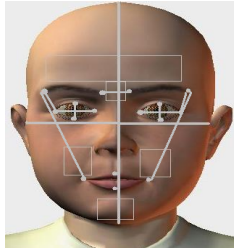
<i>Variations between Facial Expressions:</i>	
<b>Smile</b>	<b>Bored-Sleepy</b>
 <ul style="list-style-type: none"> <li>• Bigger-broader mouth</li> <li>• Slightly narrower eyes</li> <li>• Changes in the texture of the cheeks</li> <li>• Occasionally, changes in the orientation of brows</li> </ul>	 <ul style="list-style-type: none"> <li>• Head slightly turned downwards</li> <li>• Eyes slightly closed</li> <li>• Occasionally, wrinkles formed in the forehead and different direction of the brows</li> </ul>
<b>Surprise</b>	<b>Sad</b>
 <ul style="list-style-type: none"> <li>• Longer head</li> <li>• Bigger-wider eyes</li> <li>• Open mouth</li> <li>• Wrinkles in the forehead (changes in the texture)</li> <li>• Changes in the orientation of eyebrows (the eyebrows are raised)</li> </ul>	 <ul style="list-style-type: none"> <li>• Changes in the direction of the mouth</li> <li>• Wrinkles formed on the chin (different texture)</li> <li>• Occasionally, wrinkles formed in the forehead and different direction of the brows</li> </ul>
<b>Angry</b>	<b>Disgust-Disapproval</b>
 <ul style="list-style-type: none"> <li>• Wrinkles between the eyebrows (different textures)</li> <li>• Smaller eyes</li> <li>• Wrinkles in the chin</li> <li>• The mouth is tight</li> <li>• Occasionally, wrinkles over the eyebrows, in the forehead</li> </ul>	 <ul style="list-style-type: none"> <li>• The distance between the nostrils and the eyes is shortened</li> <li>• Wrinkles between the eyebrows and on the nose</li> <li>• Wrinkles formed on the chin and the cheeks</li> </ul>

Specifically, the feature extraction algorithm works as follows:

1. Search the binary face image and extract its parts (eyes, mouth and brows) into a new image of the same dimensions and coordinates as the original image.
2. In each image of a face part, locate corner points using relationships between neighboring pixel values. This results in the determination of 18 facial points, which are subsequently used to form the classification feature vector.
3. Based on these corner points, extract the specific regions of the faces (e.g. forehead, region between the eyebrows). The extracted corner points and regions can be seen in the third column in Table 3 in the Results Section, as they correspond to the six facial expressions of the same person shown in the first column. Although these regions are located in the binary face image, their texture

measurement is computed from the corresponding region of the detected face image ('window pattern') in the second column.

4. Compute the Euclidean distances between these points, depicted with gray lines in Fig. 1, and certain specific ratios of these distances. Compute the orientation of the brows and the mouth. Finally, compute a measure of the texture for each of the specific regions based on the texture of the corresponding "neutral" expression.
5. The results of the previous steps form the feature vector, which is fed into a neural network.









**Fig. 1.** The extracted features (*gray points*), the measured dimensions (*gray lines*) and the regions (*orthogonals*) of the face















### 2.3 Discriminating Features for Facial Expressions

For the classification task, we gathered 1400 images of the seven facial expressions "neutral", "smile", "sad", "surprise", "angry", "disgust" and "bored-sleepy" formed by 200 different subjects. Typical classifying features, such as, for example, the texture changes in the forehead region between the eyebrows, are shown in Table 3 for each of the seven expressions respectively.

**Table 2.** Different measures of the facial expressions

<i>Measures of texture</i>				
<b>Region between the brows:</b>				
<i>Expressions</i>	<i>Input Image</i>	<i>Difference from the relevant 'neutral expression'</i>	<i>Texture Measure</i>	<i>Possible facial expression class</i>
Neutral			0	'neutral'
Smile			16	'neutral', 'smile', 'surprise'
Surprise			6	'neutral', 'smile', 'surprise'

**Table 2.** (continued)

Angry			44	'angry', 'disgust'
Disgust			175	'angry', 'disgust'
<b>Forehead:</b>				
Neutral			0	'neutral', 'smile', 'angry', 'disgust'
Smile			0	'neutral', 'smile', 'angry', 'disgust'
Surprise			8	'surprise', 'angry'
Angry			0	'neutral', 'smile', 'angry', 'disgust'
Disgust			0	'neutral', 'smile', 'angry', 'disgust'

### 3 Results

After computing the feature vector, we use it as input to an artificial neural network to classify facial images according to the expression they contain. Some of the results obtained by our neural network can be seen in Table 3. Specifically, in the first column we see a typical input image, whereas in the second column we see the results of the Face Detection Subsystem. The extracted features are shown in the third column and finally the Facial Expression Classification Subsystem's response is shown in the fourth column.

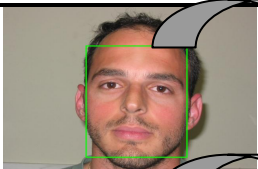


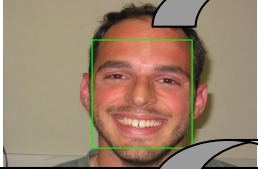
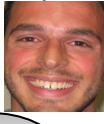

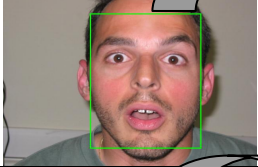
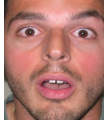

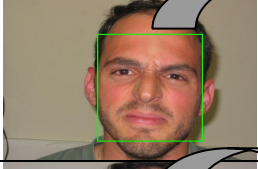
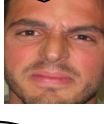

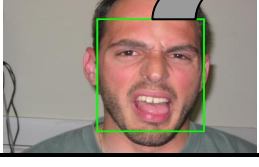
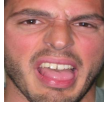

According to the requirements set, when the window pattern represented a 'neutral' facial expression, the neural network should produce an output value of [1.00; 0.00; 0.00; 0.00; 0.00] or so. Similarly, for the "smile" expression, the output must be [0.00; 1.00; 0.00; 0.00; 0.00] and so on for the other expressions. The output value can be regarded as the degree of membership of the face image in each of the 'neutral', 'smile', 'surprise', 'angry', 'disgust-disapproval' and 'bored-sleepy' classes.

### 4 Conclusions

Automatic face detection and expression classification in images is a prerequisite in the development of novel human control interactive systems. However, the development of integrated, fully operational such detection/classification systems is known to be non-trivial, a fact that was corroborated by our own statistical results regarding expression classification by humans. Towards building such systems, we

developed a neural network-based system, called NEU-FACES, which first determines whether or not there are any faces in given images and, if so, returns the location and extent of each face. Next, we described features which allow the classification of several facial expressions and presented neural network-based classifiers which use them. The proposed system is fully functional and integrated, in that it consists of modules which capture face images, estimate the location and extent of faces, and classify facial expressions. Therefore, the present or improved versions of our system could be incorporated into advanced human control interactive systems.

**Table 3.** Face Detection and Feature Extraction

	Input Image	Detected Face	Extracted Features	Expression Classification
Neutral				[ <u>1.00</u> ; 0.00; 0.00; 0.00; 0.00]
Smile				[0.16; <u>0.83</u> ; 0.00; 0.01; 0.00]
Surprise				[0.01; 0.02; <u>0.92</u> ; 0.05; 0.00]
Angry				[0.00; 0.00; 0.11; <u>0.63</u> ; 0.36]
Disgust				[0.00; 0.00; 0.08; 0.28; <u>0.64</u> ]

## 5 Future Work

In the future, we will extend this work in the following three directions: (1) we will improve our system by using wider training sets so as to cover a wider range of poses

and cases of low quality of images. (2) We will investigate the need for classifying into more than the currently available facial expressions, so as to obtain more accurate estimates of a computer user's psychological state. In turn, this may require the extraction and tracing of additional facial points and corresponding features. (3) We plan to apply our system for the expansion of human control interactive systems, in which the quality of the input images is too low for existing systems to operate reliably.

Another extension of the present work of longer term interest will address several problems of ambiguity concerning the emotional meaning of facial expressions by processing contextual information that a multi-modal human control interactive systems may provide. For example, complementary research projects are being developed [15-17] that address the problem of emotion perception of users through their actions (mouse, keyboard, commands, system feedback) and through voice words. This and other related work will be presented on future occasions

**Acknowledgements.** This work has been sponsored by the General Secretary of Research and Technology of the Greek Ministry of Development as part of the PENED-2003 basic research program.

## References

1. I.-O. Stathopoulou and G.A. Tsihrintzis, Facial Expression Classification: Specifying Requirements for an Automated System, 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems, Bournemouth, United Kingdom, October 9-11, 2006
2. P. Ekman and W. Friesen, "Unmasking the face: A Guide to Recognizing Emotions from Facial Expressions", Englewood Cliffs, NJ: Prentice Hall (1975)
3. D. Terzopoulos and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models", IEEE Transactions on Pattern Analysis and Machine Intelligence 15(6):569-579 (1993)
4. I. Essa and A. Pentland, "Coding, analysis, interpretation and recognition of facial expressions", IEEE Pattern Analysis and Machine Intelligence 19(7):757-763 (1997)
5. M.J. Black and Y. Yacoob, "Recognizing facial expressions under rigid and non-rigid facial motions", In Proceedings of the International Workshop on Automatic Face and Gesture Recognition, IEEE Press, 12-17 (1995)
6. C.L. Lisetti and D.J. Schiano, "Automatic Facial Expression Interpretation: Where Human-Computer Interaction, Artificial Intelligence and Cognitive Science Intersect", Pragmatics and Cognition (Special Issue on Facial Information Processing: Multidisciplinary Perspective), Vol. 8(1): 185-235, 2000.
7. M.N. Dailey, G.W. Cottrell, and R. Adolphs, "A six-unit network is all you need to discover happiness", in Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society, Erlbaum, Mahwah NJ, pp. 101-106, 2000.
8. M. Rosenblum, Y. Yacoob, and L. Davis, "Human expression recognition from motion using a radial basis function network architecture", IEEE Transactions on Neural Networks 7(5): 1121-1138 (1996)
9. I.-O. Stathopoulou and G.A. Tsihrintzis, "A new neural network-based method for face detection in images and applications in bioinformatics", Proceedings of the 6th International Workshop on Mathematical Methods in Scattering Theory and Biomedical Engineering, September 17-21, 2003



10. I.-O. Stathopoulou and G.A. Tsihrintzis, "A neural network-based facial analysis system," 5th International Workshop on Image Analysis for Multimedia Interactive Services, Lisboa, Portugal, April 21-23, 2004
11. I.-O. Stathopoulou and G.A. Tsihrintzis, "An Improved Neural Network-Based Face Detection and Facial Expression Classification System," IEEE International Conference on Systems, Man, and Cybernetics 2004, The Hague, The Netherlands, October 10-13, 2004.
12. I.-O. Stathopoulou and G.A. Tsihrintzis, "Pre-processing and expression classification in low quality face images", 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, Smolenice, Slovak Republic, June 29 – July 2, 2005
13. I.-O. Stathopoulou and G.A. Tsihrintzis, Evaluation of the Discrimination Power of Features Extracted from 2-D and 3-D Facial Images for Facial Expression Analysis, 13th European Signal Processing Conference, Antalya, Turkey, September 4-8, 2005
14. I.-O. Stathopoulou and G.A. Tsihrintzis, Detection and Expression Classification Systems for Face Images (FADECS), 2005 IEEE Workshop on Signal Processing Systems (SiPS'05), Athens, Greece, November 2 – 4, 2005
15. M. Virvou and E. Alepis, "Mobile educational features in authoring tools for personalised tutoring", In the journal *Computers & Education*, Elsevier, to appear (2004).
16. M. Virvou and G. Katsionis, "Relating Error Diagnosis and Performance Characteristics for Affect Perception and Empathy in an Educational Software Application," Proceedings of the 10th International Conference on Human Computer Interaction (HCI) 2003, June 22-27 2003, Crete, Greece
17. M. Virvou and E. Alepis, "Creating tutoring characters through a Web-based authoring tool for educational software," Proceedings of the IEEE International Conference on Systems Man & Cybernetics, 2003, Washington D.C., U.S.A.

# GA-Based Iris/Sclera Boundary Detection for Biometric Iris Identification

Tatiana Tambouratzis<sup>1</sup> and Michael Masouris<sup>1,2</sup>

<sup>1</sup> Department of Industrial Management and Technology, University of Piraeus,  
107 Deligiorgi St., Piraeus 185 34, Athens, Greece  
tatianatambouratzis@gmail.com

<http://www.tex.unipi.gr/dep/tambouratzis/main.htm>

<sup>2</sup> Institute of Nuclear Technology – Radiation Protection, NSCR ‘Demokritos’,  
Aghia Paraskevi 153 10, Athens, Greece  
mmasouris@kean.gr

**Abstract.** Iris identification (IRI) constitutes an increasingly accepted methodology of biometrics. IRI is based on the successful encoding and matching of distinctive iris features (folds, freckles etc.), which - in turn - presupposes that iris segmentation has been accurately performed. In contrast to the inner (iris/pupil) iris boundary, which – owing to the high contrast between the adjacent areas - is relatively easy to localize, detection of the outer (iris/sclera) iris boundary is more challenging since the low contrast between the separated areas often results in fragmented, ambiguous and spurious edges. A novel approach to iris boundary detection is presented here, featuring a genetic algorithm (GA) for outer iris boundary detection.

## 1 Introduction

Iris identification (IRI) constitutes an increasingly accepted biometrics methodology of information technology, especially as it combines a number of advantages such as non-invasiveness<sup>1</sup>, ease/speed of capture, high discriminative power, constancy over a variety of factors (aging, mood, fatigue etc.), small false acceptance and rejection rates etc. IRI is based on the successful encoding and matching of distinctive iris features (folds, freckles etc.), which - in turn - presupposes that iris segmentation has been accurately performed. In this piece of research, the iris segmentation stage preceding iris encoding and matching from near-infrared filtered images is described. Iris segmentation comprises the extraction of the iris/pupil (inner) and the iris/sclera (outer) boundaries, both of which are approximated by (not necessarily concentric) circles.

Contrary to a limited number of reported approaches [1-2], where outer iris boundary detection precedes that of the inner iris boundary, in this piece of research

---

<sup>1</sup> The equipment used for capturing the subject’s iris does not come into direct contact with the subject (e.g. compared with fingerprint biometrics); furthermore the capture conditions are such as not to make the subject feel uncomfortable, uneasy or fatigued (e.g. compared with retinal scan biometrics).

(and similar to [3-10]) inner iris boundary detection is performed first and the extracted inner boundary circle (IBC) characteristics are used for guiding outer boundary circle (OBC) localization. Such a sequence of circle finding has been preferred since - owing to the high contrast between the adjacent areas - the IBC is relatively easy to localize; on the contrary (and due to of the lower contrast between iris and sclera), OBC detection is significantly more challenging and often results in fragmented, ambiguous and spurious edges.

IBC and OBC extraction have been implemented so as to simultaneously accommodate the four well-known benchmark iris databases (namely CASIA v1.0 and CASIA v2.0 [11], MMU [12] and the Bath University Sample Database [13]), thus endowing the proposed approach with portability and universality. The images of the aforementioned databases are gray-scale images showing the iris of human subjects together with most (or all) of the eye, eyelids, eyelashes and some skin. These images have been captured employing near-infrared filters, an especially useful tool for accurately capturing the iris features of highly pigmented irises.

## 2 Inner Iris Boundary Detection

Integro-differential operators (IDO), [14], the Hough transform (HT) [15], Canny edge detection (CED) [16] and intensity thresholding (IT) constitute the main tools employed for determining the iris/pupil boundary (e.g. purely IT-based methodologies [1,7-8], IT followed by edge detection and HT [3-4] or by IDO and HT [6], CED combined with HT and IT [5,9]). Approximation of the inner iris boundary by a circle is the norm [1-10,14]; the more computationally complex approach of ellipse approximation has been implemented in [17], without - however - a significant improvement in accuracy.

Owing to the significant contrast between iris and pupil as well as the high concentration of (almost) black pixels in the latter, an IBC is quite straightforward to construct. A purely geometric approach has been implemented here<sup>2</sup> which comprises two stages: image thresholding and extraction of the IBC characteristics (centre and radius). For the following, only the central 64% part of the image has been considered (window extending above, below, to the left and to the right of the central pixel by 40% relative to the image height and length)<sup>3</sup>.

### 2.1 Image Thresholding

A threshold  $\theta$  is sought such that, after window thresholding, a black/white image pixel (with intensity lower/higher than  $\theta$ ) most probably belongs/does not belong to the pupil. Determining such a  $\theta$  is far from straightforward for IRI: on the one hand, the conditions of iris image capture (e.g. from different databases) result in different image intensity characteristics; on the other hand, even under the same conditions (e.g. within the same database), a significant variability in intensity is observed,

<sup>2</sup> This has been selected due to its ease and speed of implementation.

<sup>3</sup> Successful IRI cannot be guaranteed when the pupil lies outside this part of the image; additionally, the outer part of the image offers no pupil pixel intensity information.

which is caused – among other factors - by occlusions and light reflections on the iris/pupil/ sclera/lower eyelids etc.

Assuming that  $I_i$  ( $i=0, 1, \dots, 255$ ) are the possible gray-scale image pixel intensities<sup>4</sup> and that the window comprises  $n$  pixels, the following image characteristics have been employed for  $\theta$  calculation:

- $h_i, i=0, 1, \dots, 255$ , denoting the number of pixels of intensity  $I_i$  (obviously  $\sum_{i=0}^{255} h_i = n$ ),
- $w_i, i=3, 4, \dots, 252$ , defined as  $w_i = \sum_{k=-3}^3 h_{i+k}$  and denoting the number of pixels of intensities “around”  $I_i$ , i.e. between  $I_{i-3}$  and  $I_{i+3}$ ,
- Minimum Pixel Intensity (*MPI*), defined as  $\min_{0 \leq i \leq 255} \{I_i : h_i \neq 0\}$ ,
- Maximum Pixel Intensity (*MAPI*), defined as  $\max_{0 \leq i \leq 255} \{I_i : h_i \neq 0\}$ ,
- Average Pixel Intensity (*API*) defined as  $\frac{1}{n} \sum_{i=0}^{255} I_i h_i = \frac{1}{n} \sum_{i=MPI}^{MAPI} I_i h_i$ ,
- Maximum Intensity Area (*MIA*), defined as the pixel intensity  $I_i$  such that  $\max_{0 \leq i \leq 255} \{w_i\}$ , and denoting the central pixel intensity of the most frequently occurring consecutive pixel intensities.

These characteristics have been combined into:

$$\theta = \begin{cases} 0.34API + 0.6MPI & \text{if } MIA < 105 \\ 0.28API + 0.8MPI & \text{if } MIA \geq 105 \end{cases} \quad (1)$$

Equation (1) constitutes the first (to the authors’ knowledge<sup>5</sup>) formal expression of iris image thresholding and provides optimal  $\theta$  estimation for the iris images of the four benchmark databases. Although (1) is expected to also perform well for other iris databases, some minor coefficient adjustments may be necessary if completely different image capture conditions have been employed.

## 2.2 IBC Calculation

The IBC characteristics (centre  $(X_{IBC}, Y_{IBC})$  and radius  $R_{IBC}$ ) best approximating the pupil/iris circular boundary have been determined.

Initially,  $X_{IBC}$  is estimated, while rough estimates of  $Y_{IBC}$  and  $R_{IBC}$  are concurrently produced. To this end, 66 horizontal lines are drawn within the window, one central line passing from the central pixel of the image and 30/35 lines passing above/below the central line and spaced 1% of the image height above/below the previous line<sup>6</sup>. For each line, the line segment containing the maximum number of continuous black pixels and the location ( $X$  and  $Y$  co-ordinates) of the central pixel within the segment are collected. Subsequently, the 66 central pixels are clustered by grouping together

<sup>4</sup> Intensity  $I_0=0$  represents a black pixel, intensity  $I_{255}=255$  represents a white pixel, and increasing intensity values between 1 and 254 represent progressively lighter gray pixels

<sup>5</sup> The use of experimentally devised (database-dependent)  $\theta$  values has been mentioned in the literature, without however explicitly stating any value or formula.

<sup>6</sup> The focus is on the lower part of the window, where no occlusions of the pupil by eyelids and/or eyelashes have been observed (for the four databases).

those central pixels whose  $X$  co-ordinate values do not differ by more than a proximity value<sup>7</sup>. The largest cluster is extracted and,

- (a)  $X_{IBC}$  is selected as the most common  $X$  co-ordinate value within the cluster,
- (b) the median of the corresponding  $Y$  co-ordinate values provides a first rough  $Y_{IBC}$  estimate, and
- (c) a first estimate of  $R_{IBC}$  is given as half the length of the line segment containing the maximum number of continuous black pixels<sup>8</sup> within the cluster.

Subsequently, a finer  $Y_{IBC}$  estimate is calculated by a procedure similar to that for  $X_{IBC}$  estimation. Eleven vertical lines are drawn within the window, one central line passing from  $X_{IBC}$  and the rough estimate of  $Y_{IBC}$  and five/five lines passing to the left/right of the central line and spaced  $R_{IBC}/6$  pixels to the left/right of the previous line (i.e. located well within the pupil). For each line, the line segment containing the maximum number of continuous black pixels and the  $Y$  co-ordinate of the central pixel within this line segment are collected. The 11 central pixels are grouped by a procedure identical to that described above for the  $X_{IBC}$  estimate, whereby a finer  $Y_{IBC}$  is produced.

A finer estimate of  $R_{IBC}$  is now also possible. To this end, nine lines originating from pixel  $(X_{IBC}, Y_{IBC})$ , of orientations 0, 22.5, 157.5, 180, 202.5, 225, 270, 315 and 337.5° and of length  $R_{IBC}$  (rough estimate) are drawn. Subsequently, the number of continuous black pixels on each line is counted; each number is then converted into an actual segment length. The majority of the nine segment lengths is expected to be close<sup>9</sup> to the initial  $R_{IBC}$  estimate; after any irrelevant lengths (that may appear due to light reflections or shadows in the selected directions) have been excluded, the average of the remaining lengths constitutes the finer  $R_{IBC}$  estimate.

Following IBC extraction, the validity of the IBC characteristics is verified: two further circles - concentric to IBC but with radii equaling  $0.70 R_{IBC}$  and  $1.30 R_{IBC}$  - are created and the percentage of black image pixels between the IBC and each of them is counted. The IBC is accepted if the area between the IBC and the smaller circle contains more than 80% black pixels and the area between the IBC and the larger circle contains less than 20% black pixels. Failure to satisfy this criterion<sup>10</sup> necessitates re-estimation of  $Y_{IBC}$  and  $R_{IBC}$  via a series of  $Y$ - $R$  tests. For each test,

- A circle with centre  $(X_{IBC}, Y)$  and radius  $R$  is constructed, where  $Y$  and  $R$  take values close to the estimated  $Y_{IBC}$  and  $R_{IBC}$ <sup>11</sup>, respectively.
- Circle evaluation is performed by drawing two concentric circles with radii  $(R-2)$  and  $(R+2)$  and calculating the percentage of black pixels in the areas between the constructed circle and the concentric circles.

IBC selection corresponds to the circle with the higher percentage of black pixels in the inner area and, if equal, that with the lower percentage in the outer area. The

<sup>7</sup> The proximity value is directly associated with the image dimensions and equals  $\{0.5\%$  of image width + 1 $\}$ .

<sup>8</sup> If the selected estimate exceeds the range of the  $Y$  co-ordinate values observed in the cluster, the next smaller  $Y$  co-ordinate value of (b) is employed.

<sup>9</sup> Closeness defined by the previously mentioned proximity value (see footnote 7).

<sup>10</sup> When the pupil is heavily occluded by eyelids, eyelashes and/or shadows, the measured numbers either may not include the entire pupil or may extend outside it.

<sup>11</sup> With decrements of one pixel in the value of  $Y$  per step.

results concerning the quality of IBC detection, namely the proportion of images with successfully extracted IBC, the proportion of images requiring final  $Y_{IBC}$  and  $R_{IBC}$  re-estimation, the average number of  $Y$ - $R$  steps required until IBC selection and the average execution time are presented in Table 1. It is worth pointing out that the 1% failure rate in the MMU database is due either to the pupil being heavily occluded by eyelids and/or light-colored eyelashes or to a multitude of spurious reflections appearing on the pupil (failure during image capture).

**Table 1.** IBC detection results

Database	Success rate (%)	Need for $Y$ - $R$ tests (%)	Required $Y$ - $R$ Steps <sup>12</sup>	Execution time (ms) <sup>13</sup>
CASIA v2.0	100	0	0	1
CASIA v2.0	100	17	110	12
MMU	99	10	56	3
Bath	100	5	620	26

### 3 Outer Iris Boundary Detection

As for IBC, IDO [6,8,14] and CED combined with HT [2-5,9-10] have been employed for OBC extraction. Although the CED-HT combination has been found especially successful in circle detection, its main drawback is high computational demands.

Taking into account that the (normal) pupil diameter ranges between 3 and 8mm<sup>14</sup>, the (normal) iris diameter is around 12mm long [20], and the OBC centre lies within the pupil<sup>15</sup>, it is only necessary to consider the square window extending  $4.5R_{IBC}$  above, below, to the left and right of  $(X_{IBC}, Y_{IBC})$ . Two stages have been preformed: a pre-processing (circular OBC edge extraction) stage and a genetic algorithm (GA) for determining the OBC characteristics (centre and radius).

#### 3.1 Window Pre-processing

Pre-processing aims at:

- extracting as many circular edges as possible which, most probably, constitute parts of the OBC circumference,
- removing circular edges that contribute to false detection.

<sup>12</sup> The average number of  $Y$ - $R$  steps required is highly dependent on the size of the original iris image.

<sup>13</sup> The algorithm described here has been implemented in JAVA; the use of a more efficient language (e.g. C++) should further speed up the reported execution times.

<sup>14</sup> In pathological situations or under extreme lighting conditions (which are not of interest here), the pupil diameter can be as small as 1.5mm [18-19].

<sup>15</sup> It has been found that, for the four databases, the OBC centre lies within the circle with centre that of the IBC and radius  $0.34R_{IBC}$ .

In order to compensate for the relatively low iris/sclera contrast, pre-processing involves scaling, smoothing and contrast enhancement, followed by edge detection and filtering.

Window scaling has been performed since a high resolution is not necessary for iris segmentation. The amount of scaling is based on the image dimensions (here, windows of width between 100 and 160 pixels are produced) and aims at (a) reducing the computational effort, (b) maximally suppressing the local luminosity changes, and (c) not affecting edge detection accuracy. Subsequently, a Gaussian Smoothing Filter [21] has been applied to the scaled window in order to eliminate sparse noise from irrelevant sources (eyelashes, skin, light reflections etc); the amount of smoothing is determined by the selected standard deviation of the Gaussian and the size of the convolution mask. Finally, contrast stretching (normalisation) [21] has been employed in order to enhance the iris/sclera contrast.

Owing to its capability of outputting fine connected curves<sup>16</sup>, CED has been preferred among a wide variety of feature detectors (Canny, Sobel, Roberts, Sarkar-Bowyer etc. [16,21-22]). A smoothing step is initially applied, which produces a slightly blurred version of the window. Subsequently, the edges along each direction are tracked and non-maxima are suppressed via a map of intensity gradients resulting into thin curves. Finally, the appropriate selection of high and low threshold values reveals the edges that possibly lie on the iris/sclera boundary, while hysteresis thresholding prevents noisy edges from being broken into multiple edge fragments.

Edge filtering eliminates the previously revealed edges that do not contribute to the OBC circumference, e.g. edges appearing:

- on the upper or lower parts of the iris and corresponding to eyelids, eyelashes and/or light reflections (to this end, elimination of the exposed edges lying between 30 and 150° as well as between 255 and 285° relative to point  $(X_{IBC}, Y_{IBC})$  has been performed),
- too near the IBC circumference;
- on the circle with centre  $(X_{IBC}, Y_{IBC})$  and radius  $1.5R_{IBC}$ .

### 3.2 OBC Calculation

Genetic algorithms (GAs) [23-25] constitute a relatively novel stochastic exploration methodology which is capable of quickly reaching optimal (or near-optimal) solutions to a variety of problems. GAs are inspired by such aspects of natural evolution as inheritance, selection, mutation, crossover. They operate as follows: a set of candidate solutions (population of chromosomes) is created and updated stochastically (in a number of generations) according to the quality of each candidate solution (chromosome fitness) until either an optimal solution (chromosome of maximum fitness) is found, the maximum allowable number of updates is reached or some other termination criterion is satisfied. Stochastic updating of the set of candidate solutions promotes the creation of progressively better solutions (chromosomes of higher fitness) in the set, whereby it becomes increasingly probable - at each subsequent update - for a (near-)optimal solution to be found in the population.

<sup>16</sup> Unlike other edge detectors that may produce multiple curved edges as a response to single edges or to broken/disconnected edges.

Shape-detection GAs appear in the literature for circles, ellipses, squares, rectangles etc. [24-26]. The proposed GA employs three-gene chromosomes of the form  $(X_{OBC}, Y_{OBC}, R_{OBC})$  for (near-)optimal OBC circle detection<sup>17</sup>. Chromosome fitness is expressed via the normalized<sup>18</sup> total number of pixels of the candidate OBC that belong to a circular edge extracted during the pre-processing stage; the maximum fitness value reaches about 0.58, which is due to the fact that edges appearing on the top (33.3%) and bottom (8.4%) of circle have been filtered.

The population size has been set to 60. For the first generation - and in order to speed up convergence -, directed chromosome construction has been implemented: 120 chromosomes are randomly created and, from these, the 60 chromosomes of highest fitness are included in the initial population. The termination criterion combines a maximum number of 35 generations or a fitness value of no less than 0.52. Generation evolution comprises:

- One-point circular crossover between pairs of chromosomes. Sixty crossovers have been performed on pairs of chromosomes selected via roulette-wheel according to their fitness, resulting into 120 offspring. A selected pair of chromosomes  $(X^1_{OBC}, Y^1_{OBC}, R^1_{OBC})$  and  $(X^2_{OBC}, Y^2_{OBC}, R^2_{OBC})$  may combine equi-probably into one of the following pairs of offspring  $(X^2_{OBC}, Y^1_{OBC}, R^1_{OBC})$  and  $(X^1_{OBC}, Y^2_{OBC}, R^2_{OBC})$ ,  $(X^1_{OBC}, Y^2_{OBC}, R^1_{OBC})$  and  $(X^2_{OBC}, Y^1_{OBC}, R^2_{OBC})$ , or  $(X^1_{OBC}, Y^1_{OBC}, R^2_{OBC})$  and  $(X^2_{OBC}, Y^2_{OBC}, R^1_{OBC})$ .
- Random mutation of each gene of a chromosome with a mutation probability of  $p^{mut}=0.3$ ; the mutated gene value  $Z^{mut}$  (where  $Z$  can take on values  $X, Y$  or  $R$ ) is given by

$$Z^{mut}_{OBC} = \alpha Z_{OBC} \quad (2)$$

where  $Z_{OBC}$  the original gene value and  $\alpha$  is a uniformly selected number from [0.985, 1.015].

- Multi-purpose selection that maintains the diversity of the population while preventing premature convergence at local optima. The population of the new generation comprises the 52 fittest (out of the 120 created) offspring, the fittest chromosome of the previous generation, and seven randomly created new chromosomes. A check for identical chromosomes is performed and, if such chromosomes are found, all but one of them are mutated according to (2) with  $p^{mut}=1$  per gene.

### 3.3 OBC Detection Results

The accuracy and computational complexity of the proposed OBC detection approach has been evaluated on the four benchmark databasets; to this end, the pixel intensity inside and outside the OBC circumference has been compared.

<sup>17</sup>  $(X_{OBC}, Y_{OBC})$  is the centre and  $R_{OBC}$  the radius of the candidate OBC. The alternative chromosome encoding via triplets of points on the OBC circumference [24], has not been followed due to problems arising when pairs of points are very near to each other or when restrictions concerning the centre and/or radius of the OBC are not met.

<sup>18</sup> Normalization involves dividing by the number of pixels in the circumference of the candidate OBC, performed in order to accommodate for OBCs of different radii.



Furthermore, a comparison with the HT for OBC detection – considering both the original image and the selected window only - has been performed. To this end, a three-dimensional array, constructed for storing centres and radii of the possible OBC circles, has been initialized to zero. During operation, and for each image point detected by CED, every element of the array corresponding to a circle center and radius  $R$  that fits the image point is increased by  $1/R$ <sup>19</sup>. After operation, the element with the maximum value corresponds to the centre and radius of the OBC.

**Table 2.** Comparison of OBC detection success rates (%) and average execution times (ms)

Database	GA-based		HT (entire image)		HT (window only)	
	success	efficiency	success	efficiency	success	efficiency
CASIA v1.0	95	141	87	145	93	140
CASIA v2.0	97	149	90	124	95	115
MMU	99	153	98	157	98	97
BATH	97	128	94	135	97	121

The HT has been applied to all the edges extracted by CED as well as to only those edges that belong to the window extending  $4.5R_{IBC}$  above, below, to the left and right of  $(X_{IBC}, Y_{IBC})$ . Table 2 illustrates that the GA-based approach is superior to HT in approximating the outer iris boundary, especially for the two CASIA databases; the advantages of the proposed approach become more accentuated when HT is applied to the entire image.

## 4 Conclusions

Successful and portable iris segmentation (inner and outer iris boundary detection) has been demonstrated on the iris images of the four iris benchmark databases. The proposed geometric approach to inner iris boundary detection is accurate, straightforward and fast; inner iris boundary detection has been implemented so as to precede the detection of the outer iris boundary since it limits the complexity of the latter. A novel genetic algorithm-based approach to outer iris boundary extraction has been found superior to existing image processing techniques, both in terms of accuracy and in terms of computational demands. Finally, the advantages of restricting both inner and outer iris boundary detection to appropriately selected image windows have been exploited.

**Acknowledgements.** This research has been carried out in the framework of the European Union Project for the Reinforcement of Scientific Potential<sup>20</sup> entitled “**Biometric Automated Identification: Iris Recognition for the Selective Access to Areas of Increased Security**”.

<sup>19</sup> The involvement of  $R$  in normalizing the increment is necessary in order to compensate for the disproportionately higher scores attained for candidate circles of larger  $R$  values.

<sup>20</sup> ΠΕΝΕΔ’03 Programme 03ΕΔ504, co-funded by the European Union and the Greek Secretariat for Research and Technology.

## References

1. Liam, L., Chekima, A., Fan, L., Dargham, J.: Iris Recognition Using Self-Organizing Neural Networks. IEEE 2002 Student Conference on Research and Developing Systems, Malaysia (2002) 169-172
2. Masek, L.: Recognition of Human Iris Patterns for Biometric Identification, BSc Thesis, Univ. Western Australia (2003)
3. Wildes, R., Asmuth, J., Green, G., Hsu, S., Kolczynski, R., Matey, J., McBride, S.: A Machine-Vision System for Iris Recognition. *Machine Vision and Applications* **9** (1996) 1-8
4. Wildes, R.: Iris Recognition: An Emerging Biometric Technology. *Proceedings of the IEEE* **85** (1997) 1348-1363
5. Ma, L., Tan, T., Wang, Y., Zhang, D.: Personal Identification Based on Iris Texture Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25** (2003) 1519-1533
6. Tisse, C., Martin L., Torres, L., Robert, M.: Person Identification Technique Using Human Iris Recognition. *St Journal of System Research* **0** (2002) 92-100
7. Maenpaa, T.: An Iterative Algorithm for Fast Iris Detection. International Workshop on Biometric Recognition Systems 2005, Beijing, China. *Lecture Notes in Computer Science*, Springer-Verlag, Berlin (2005) 127-134
8. Teo, C., Ewe, H.: An Efficient One-Dimensional Fractal Analysis for Iris Recognition. Short Paper Proceedings of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG'2005, Plzen-Bory, Czech Republic (2005) 157-160
9. Liu, X., Bowyer, K., Flynn, P.: Experiments with an Improved Iris Segmentation Algorithm. *Automatic Identification Advanced Technologies 2005*, Fourth IEEE Workshop, Buffalo, New York, USA (2005) 118-123
10. Cui, J., Wang, Y., Tan, T., Ma, L., Sun, Z.: An Iris Recognition Algorithm Using Local Extreme Points. *International Conference on Biometric Authentication*, Hong Kong (2004) 442-449
11. CASIA Iris Database <http://www.sinobiometrics.com/>
12. Multimedia University - Malaysia (MMU) Iris Database
13. Bath University Iris Database, <http://www.bath.ac.uk/eleceng/pages/sipg/irisweb/database.html>
14. Daugman, J.: How Iris Recognition Works. *IEEE Transactions on Circuits and Systems for Video Technology* **14** (2004) 21-30
15. Hough, P.C.V.: Machine Analysis of Bubble Chamber Pictures. *International Conference on High Energy Accelerators and Instrumentation*, CERN, Geneva (1959) 554-556
16. Canny, J.: A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8** (1986) 679-698
17. Miyazawa, K., Ito, K., Aoki, T., Kobayashi, K., Nakajima, H.: A Phase-Based Iris Recognition Algorithm. *International Conference on Biometrics*, Hong Kong (2006) 356-365
18. Atchison, DA., Smith, G., Efron, N.: The Effect of Pupil Size on Visual Acuity in Uncorrected and Corrected Myopia. *American Journal of Optometry and Physiological Optics* **56** (1979) 315-323
19. Thibos, L., Miller, D.: Electronic Spectacles for the 21st Century. *Indiana Journal of Optometry* **2** (1999) 6-10
20. Lefohn, A., Budge, B., Shirley, P., Caruso, R., Reinhard, E.: An Ocularist's Approach to Human Iris Synthesis. *IEEE Computer Graphics and Applications* **23** (2003) 70-75

21. Vernon, D.: *Machine Vision*. Prentice-Hall, New York (1991)
22. Heath, M., Sarkar, S., Sanocki, T., Bowyer, K.: Comparison of Edge Detectors. *Computer Vision and Machine Understanding* **69** (1998) 38-54.
23. Whitley, D.: A Genetic Algorithm Tutorial. *Statistics and Computing* **4** (2003) 65-85
24. Ayala-Ramirez, V., Garcia-Capulin, C.H., Perez-Garcia, A., Sanchez-Yanez, R.E., : Circle Detection on Images Using Genetic Algorithms. *Pattern Recognition Letters* **27** (2006) 652-657
25. Yao, J., Kharna, N., Grogono, P.: A Multi-Population Genetic Algorithm for Robust and Fast Ellipse Detection. *Pattern Analysis and Application* **8** (2005) 149-162
26. Mainzer, T.: Genetic Algorithm for Traffic Sign Detection. *Proceedings of International Conference in Applied Electronics, Pilsen, University of West Bohemia* (2002) 129-132

# Neural Network Based Recognition by Using Genetic Algorithm for Feature Selection of Enhanced Fingerprints

Adem Alpaslan Altun and Novruz Allahverdi

Selcuk University, Technical Education Faculty  
Department of Computer Systems Education, 42031 Konya, Turkey  
{altun, noval}@selcuk.edu.tr

**Abstract.** In order to ensure that the performance of a fingerprint recognition system will be powerful with respect to the quality of input fingerprint images, the enhancement of fingerprints is essential. In this study wavelet transform and contourlet transform which is a new extension of the wavelet transform in two dimensions are applied for fingerprint enhancement. In addition, feature selection is a process that chooses a subset of features from the original fingerprint features so that the feature space is optimally reduced according to a certain criterion. In this study, a Genetic Algorithms (GAs) approach to fingerprint feature selection is proposed and selected features are input to Artificial Neural Networks (ANNs) for fingerprint recognition. The performance has been tested on fingerprint recognition.

## 1 Introduction

Among biometrics, fingerprint-based identification is one of the most advanced and reliable technique. The ridge structures in fingerprint images are not always well defined in the recognition phase, and therefore, the aim of the fingerprint enhancement can be summarized as recovering the topology structure of ridges and valleys from the noisy image [1]. A number of algorithms have been proposed to enhance gray level fingerprint images. Do and Vetterli [2] utilized a double filter bank structure to develop the contourlet transform and used it for some non-linear approximation and denoising experiments. In this study, we apply wavelet transform and contourlet transform for fingerprint image enhancement.

In addition, a method is proposed to match enhanced fingerprints. We use a new representation for the fingerprints which is developed by Jain et al. [3] and yields a comparatively short, fixed length code, called FingerCode suitable for texture-based matching. A feature vector, called FingerCode, is the collection of all the features for every sector in each filtered image. Consequently, the feature elements extract the local information and the ordered enumeration of the tessellation captures the invariant global relationships among the local patterns. In our study FingerCode are used fingerprint matching based on a set of ANNs.

In general, ANNs can produce robust performance when a large amount of data is available. However, it may not be possible to train ANNs or the training task cannot be effectively carried out without data reduction when a data set is too huge. These

problems are especially critical in the case of using back propagation algorithms (BP) of ANNs. Under these conditions, data reduction techniques can be achieved in many ways such as feature selection or data reduction [4], [5], [6]. Among those, GAs have proven to be an effective computational method, especially in situations where the search space is highly dimensional.

In this study, we apply a method for selection of the training data which have the highest selective power using a GAs and some methods for the recognition of fingerprints using BP algorithms such as Quick Propagation (QP), Online Back-Propagation (OBP), Batch Back-Propagation (BBP) and Conjugate Gradient Descent (CGD).

## 2 Fingerprint Image Enhancement

In order to ensure that the performance of a fingerprint recognition system will be robust referring to the quality of fingerprint images, an enhancement algorithm which will improve the clarity of the ridge/valley structures is necessary. This paper investigates the problem of fingerprint denoising when the fingerprint is corrupted by additive white Gaussian noise, which is a common situation for fingerprints obtained through scanning or other image capturing devices. In the enhancement phase, firstly the fingerprint images are contaminated by a zero-mean Gaussian noise with mean value selected as 0 and variance value selected as 0.02.

Several works on noise reduction are based on wavelet thresholding [7], a simple and very effective denoising method. We apply wavelet transform and contourlet transform which is a new extension of the wavelet transform in two dimensions for denoising of fingerprint images.

### 2.1 Wavelet Transform for Fingerprint Image Denoising

One of the most powerful transform tools in image denoising is the wavelet transform. The wavelet transform is based on a fundamental two-channel filter bank. These filters form one stage of the filter-bank structure as shown in Fig. 1. Each bank decomposes the fingerprint image into lowpass and highpass sub-bands [7]. The filter bank has a low-pass and a highpass filter, and each is followed by a 2:1 decimator. At every stage of the two-channel filter bank, the number of input samples is maintained; i.e., for  $N$  input samples, there are  $N/2$  lowpass output samples and  $N/2$  highpass ones. For the two-dimensional case, the fingerprint image is decomposed into four decimated subbands by applying analysis filter banks in each of horizontal and vertical directions as depicted in Fig. 2 for the analysis section [13].

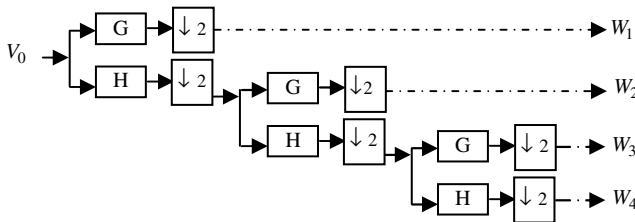
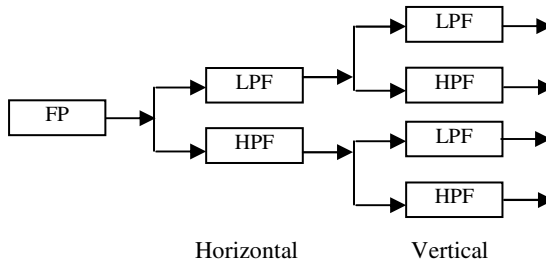


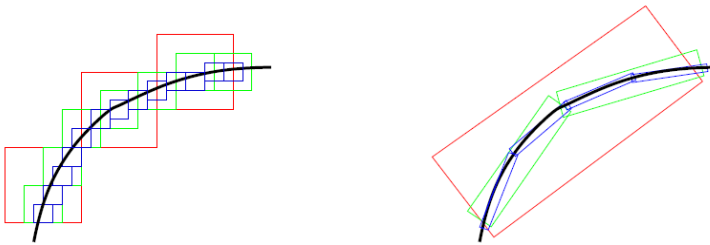
Fig. 1. Filter-bank structure implementing a discrete wavelet transforms



**Fig. 2.** Wavelet transform of a fingerprint image is based on the elementary two-channel filter bank, which, in two dimensions, decomposes the image into lowpass and highpass sub-bands in each of the horizontal and vertical directions

## 2.2 Contourlet Transform for Fingerprint Image Denoising

The contourlet transform is composed of basis images oriented at various directions in multiple scales, with flexible aspect ratios. With this rich set of basis images, the contourlet transform effectively capture smooth contours that are dominant feature in natural images [12]. Refer to [8], 2-D wavelet transforms are only good at catching point discontinuities, but don't capture the geometric smoothness of the contours. The contourlet transform was improved for the solution of this problem. Wavelet transforms are the square transforms which can describe only point-wise discontinuances. But contourlet transforms are able to creep over the linear parts of contours and therefore, less number of coefficients for the suitable describing of a continuous contour, are necessary. With such a rich set of basis functions, contourlets can represent a smooth contour with smaller number coefficients compared with wavelets, as illustrated in Fig. 3.



**Fig. 3.** Wavelets have square supports that can only capture point discontinuities. Whereas contourlets have extended supports that can capture linear segments of contours, and thus can effectively represent a smooth contour with fewer coefficients.

In this study, in order to establish an objective assessment of the system, it was tested on NIST-4 fingerprint database contained 512x512 8-bit gray-scale fingerprint images. Each filtering techniques was applied fingerprint images. We use peak signal-to-noise ratio (PSNR) as the metric for objective fingerprint image quality. The PSNR is defined as

$$PSNR = 10 \log_{10} \left\{ \frac{255^2}{\frac{1}{NM} \sum_{n_1=1}^N \sum_{n_2=1}^M [f(n_1, n_2) - \hat{f}(n_1, n_2)]^2} \right\} \quad (1)$$

where  $f(n_1, n_2)$  and  $\hat{f}(n_1, n_2)$ ,  $1 \bullet n_1 \bullet N$ ,  $1 \bullet n_2 \bullet M$ , are the original fingerprint image and the enhanced image with size  $N \times M$ , respectively.

The obtained signal-to-noise ratio results of the filtering techniques are shown in Table 1. The PSNR of the noisy fingerprint image is 12.18 dB when it's compared with the original. Comparing the fingerprint images denoised by the filtering techniques, we conclude that the perceptual quality of the enhanced fingerprint images is significantly better than the noisy ones. The PSNR between the original and the restored fingerprint image using the wavelet transform is 18.21 dB, whereas the PSNR result using the contourlet transform is 19.65 dB. Although all methods improve the quality of the distorted images, contourlet-based method achieves better performance, i.e., higher PSNR and better quality, than the other filtering techniques. We see that contourlets are superior compared with wavelets in capturing directional textures on fingerprint images.

**Table 1.** Peak Signal-Noise Ratio (in dB) of between noised and denoised fingerprint images

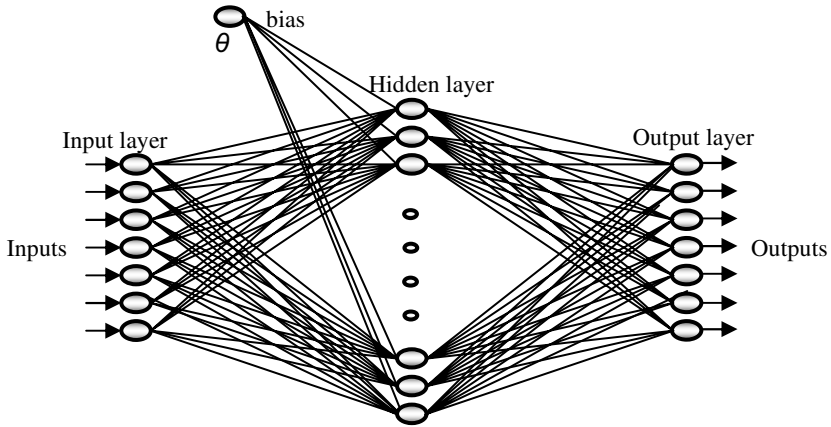
<i>Fingerprint Images</i>	<i>Peak Signal-to-Noise Ratio (dB)</i>
Noised Fingerprint Image	12.18
Restored Fingerprint Image using Wavelet Transform	18.21
<b>Restored Fingerprint Image using Contourlet Transform</b>	<b>19.65</b>

### 3 Recognition of Fingerprints Using ANNs

ANNs are the calculation model that is inspired by working principles of biological neural systems. It works as the principle of decision by learning. Multilayer perception model is a kind of feedforward ANNs which has input layer, hidden layer and output layer [9]. Each layer is fully connected to the succeeding layer, as shown in Fig. 4.

Lots of learning algorithms can be used in training of multilayer ANNs such as Back propagation algorithm. The conjugate gradient algorithm is an improved method of multilayer perception training [10]. It is suitable for the network which has huge number of output nodes and weights. In this work Quick Propagation, Online Back-Propagation, Batch Back-Propagation and Conjugate Gradient Descent are used for training of multilayered ANNs.

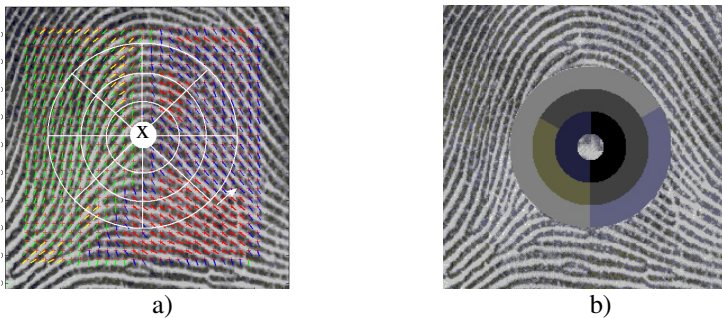
It is necessary to extract the feature vectors of fingerprints before fingerprints are recognized by using ANNs. For this reason, reference points and region of interest around the reference points are determined by obtaining directional histograms in the enhanced fingerprint images [12]. In the denoised fingerprint images, a predetermined



**Fig. 4.** A typical feedforward, fully connected network with three layers of neurons

region of interest around the reference point then is tessellated into cells, as shown in Fig. 5a. We consider different concentric bands around the detected reference point for feature extraction. Each band is different pixels wide such as 20 ( $b=20$ ), and segmented into different sectors such as 16 ( $k=16$ ). A band with a width of 20 pixels is necessary to capture a single minutia in a sector, allowing our low level features to capture this local information [11, 12].

A feature vector is composed of an ordered enumeration of the fingerprint features extracted from the local information contained in each subimage (sector) specified by the tessellation. As shown in Fig. 5b, a feature vector called FingerCode, is the collection of all the features (for every sector) in each enhanced fingerprint images [11]. These features capture both the global pattern of ridges and valleys and the local characteristics. In the fingerprint recognition process, we used feature vectors as an input of ANNs.



**Fig. 5.** a) Reference point (x), the region of interest, and 24 sectors superimposed on a fingerprint, b) A feature vector called FingerCode of the fingerprint



The topologies of ANNs which are used training and testing of the ANNs and the training effort are briefly described by the following statistics:

- Number of inputs= various (related to the band numbers and the sector numbers of FingerCode)
- Number of outputs= 10 (number of people)
- Number of hidden layers= 1
- Number of hidden neurons= various (related to the number of inputs of ANNs)
- Number of training patterns= 1000
- Learning rate= 0.9
- and Momentum= 0.1.

After all parameters had been determined, the feature vectors of fingerprints were input to ANNs for training and testing. As a result of applying ANNs, the performance of the fingerprint recognition method has been tested. Since the feature vectors trained by ANNs are huge in number, the training phase may take a long time. Thus, the feature vectors are reduced by using Genetic Algorithms (GAs):

### 4 Feature Selection of Fingerprints Using Genetic Algorithms

Genetic algorithms (GAs) introduced by Holland [5] are a well known searching algorithm in a large space. GAs are used in feature selection problems [6]. GAs employ a fixed length binary string to represent a possible solution for a problem domain. In this case, let  $L$  is a total number of features, there exist  $2^L$  possible feature subsets. Each individual is represented by an  $L$ -bit string. Value '1' or '0' of any bit means present or absent of the corresponding feature, respectively (Fig. 6).

The initial set of possible solutions or population with a fixed number of population or population size is randomly constructed. After the initialization step, each chromosome is evaluated by the fitness function. According to the value of the fitness function, the chromosomes associated with the fittest individuals will be reproduced more often than those associated unfit individuals [4]. New individuals (offspring) for the next generation are formed by using two main genetic operators, crossover and mutation. They provide the means for introducing new information into the population. Finally, the GAs tend to converge on optimal or near-optimal solutions.

The GAs are usually employed to improve the performance of artificial intelligence techniques [5]. For ANNs, the GAs were applied to the selection of ANNs topology including optimizing a relevant feature subset and determining the processing

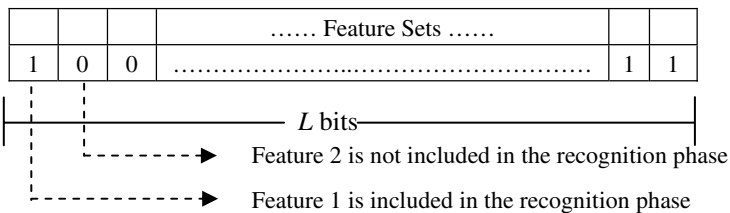


Fig. 6. An  $L$ -dimensional binary vector determined according to fingerprint feature sets

elements. GAs have been already used for feature selection using different learning algorithms to evaluate the fitness of subsets of attributes [6]. Figure 7 shows a general framework for the application of a feature selection algorithm using GAs for fingerprint images. In this study, GAs are used to improve the robustness of feature selection without sacrificing too much computational efficiency.

The GAs are applied to maximize the fitness of the best selected feature vectors. The following GA parameters are used for the selected features of fingerprints.

- Population size= 100
- Generations= 50
- Crossover probability= 0.9
- Mutation probability=0.01

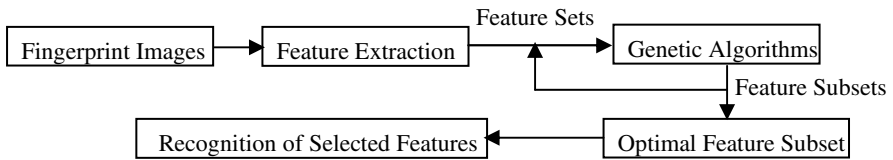


Fig. 7. Feature selection strategy using GAs

The feature selection method starts with a random population of input configuration. Input configuration determines which inputs are ignored during performance test. At each following step (called generation); a process analogous to natural selection to select superior configurations is used. Generations, then are used to generate a new population. Each step successively produces better input configuration. In 50 generations, GAs always find the global optimum in the constrained search space. At the last step, the best configuration is selected. In this study, the method is very time-consuming but good for determining mutually-required inputs and detecting interdependencies. Moreover GAs are used probabilistic neural networks (PNNs) as the fitness function because PNNs are trained quickly and proved to be sensitive to the irrelevant inputs.

After determining all parameters, the feature vectors are reduced by using GAs. Thus, feature mask of the best network is determined as a string like ‘010000011001111.....’. Here, value of ‘1’ or ‘0’ suggests that feature is selected or removed respectively. In this study, the selected feature vectors are again input to ANNs for training and used to test the performance of the fingerprint recognition method. The GAs-based method proved to be quite effective in improving the robustness of the feature selection over the feature vectors of fingerprints.

## 5 Experimental Results

The feature selection with a genetic algorithm was applied on the sets of fingerprint features called FingerCode. The genetic algorithm looked at  $N$  individuals in a generation,  $(b_1, b_2, \dots, b_N)$ , where each individual was 384, 512, 768, 1120 features vector, i.e.,  $b_i \in B^{384}$ . The GAs for the feature selection provided good results since it reduced available features almost 50%. After selection, the number of the feature vectors is 196, 238, 354 and 517, respectively.

The neural network was trained on the 700 training samples but was then validated and tested on a set of 150 samples from the validation and test sets, respectively. In addition, this 1000 sample set was also used to evaluate the fitness of the population. The population size was selected as 100. As in the experiment, the probability of crossover was selected as 0.9 and the mutation probability as 0.01. The two best individuals in a generation were kept unchanged for the next generation.

The inputs of ANNs are constituted by the fingerprint feature vectors. The output nodes extract the information that has these fingerprint images. In this work the NIST-4 fingerprint database is used. This database concerns 1000 fingerprint images which are obtained from 10 people taken in different times. 700 of these fingerprint images are used in training (TRN) process, 150 of these are used in validation (VLD) process and the rest is used in testing (TST) process. After we determined the optimum ANNs structure depending on the number of hidden layer node, to confirm the efficiency of our feature vectors, a feedforward neural network with a single hidden layer was trained with backpropagation algorithms. In Table 2 and Table 3, the correct classification rates (%CCR) according to the band and sector number, ANNs methods and

**Table 2.** The effects of the enhancement methods on the performance of ANNs for recognition of fingerprints

Enhancement Methods	Band number and sector number (ANNs structures)	ANNs Methods	Correct Classification Rate (CCR%) for datasets			
			TRN	VLD	TST	All
Contourlet	7 x 20 (1120:168:10)	BBP	99.4	98.0	96.6	98.8
Wavelet	7 x 20 (1120:168:10)	BBP	98.8	96.0	96.6	98.1
<b>Contourlet</b>	<b>4 x 12 (384:58:10)</b>	<b>CGD</b>	<b>100.0</b>	<b>98.6</b>	<b>98.6</b>	<b>99.6</b>
Wavelet	4 x 12 (384:58:10)	CGD	100.0	96.6	95.3	98.8
Contourlet	6 x 16 (768:116:10)	OBP	100.0	97.3	98.6	99.4
Wavelet	6 x 16 (768:116:10)	OBP	99.8	98.0	95.3	98.9
Contourlet	4 x 16 (512:78:10)	QBP	99.1	97.5	96.0	98.4
Wavelet	4 x 16 (512:78:10)	QBP	98.9	96.5	96.4	98.2

**Table 3.** The effects of the image enhancement methods and selected features by using GAs on the performance of ANNs for recognition of fingerprints

Enhancement Methods	Band number and sector number (ANNs structures)	ANNs Methods	Correct Classification Rate (CCR%) for datasets			
			TRN	VLD	TST	All
Contourlet	7 x 20 (517:84:10)	BBP	99.5	97.8	97	98.9
Wavelet	7 x 20 (517:84:10)	BBP	98.0	94.0	94.6	96.9
<b>Contourlet</b>	<b>4 x 12 (196:31:10)</b>	<b>CGD</b>	<b>100.0</b>	<b>99.0</b>	<b>98.7</b>	<b>99.7</b>
Wavelet	4 x 12 (196:31:10)	CGD	100.0	96.5	95.1	98.7
Contourlet	6 x 16 (354:60:10)	OBP	100.0	97.5	98.7	99.4
Wavelet	6 x 16 (354:60:10)	OBP	99.8	97.8	96.1	99.0
Contourlet	4 x 16 (238:39:10)	QBP	99.3	97.6	98.0	98.9
Wavelet	4 x 16 (238:39:10)	QBP	99.1	97.0	97.0	98.5

feature selection of GAs are shown, respectively. In Table 2, the results of unselected features are shown; but in Table 3, it is shown the results of selected features using GAs. The best results are taken after training 100 iterations of the ANNs.

From these results, the sectorization process is realized by 4 bands and 12 sectors. As a result of sectorization, 384 feature vectors are obtained from all fingerprints. It is applied ANNs methods such as CGD for recognition of fingerprints. The conjugate gradient methods is also applied to ANNs structure which has 10 node output layer, 31 node hidden layer and 196 node input layer for selected 196 feature vectors. After the training process of 100 iterations, the success rate of 100 percent is obtained both normal features and selected features using GAs for training datasets. By considering of test set, the success rate is 98.7 percent and for all of the data the success rate is 99.7 percent for the selected features. It has been observed that the result of GAs feature selected approach is more successful than the normal features.

## 6 Conclusion

One of the most important steps of matching process in fingerprint recognition is image enhancement stage. Also the resolution of fingerprint image affects the success rate of matching directly. The best performance of image enhancement is achieved contourlet transform because of suitability for arcuate structure of fingerprint. The effectiveness of fingerprint image enhancement can be increased by modifying the contourlet transform. The optimum band number can be chosen as 4 or 5 and the sector number can be chosen as 12 or 16 to obtain feature vectors. The duration of ANNs training process is high because of the huge number of feature vector which is given as an input of ANNs. The numbers of these feature vectors are decreased by using data reduction methods such as Genetic Algorithm. In this study, GAs were used to select a good subset of the feature vectors in order to improve the fingerprint recognition performance.

**Acknowledgments.** This study is supported by the department of the Scientific Research Projects of Selcuk University in Turkey.

## References

1. Hong L., Wan Y., Jain A.K., Fingerprint Image Enhancement: Algorithms and Performance Evaluation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20(8), (1998) 777-789.
2. Do M.N., Vetterli M., Contourlets: a directional multiresolution image representation, *Image Processing, International Conference*, vol. 1, (2002) 357-360.
3. Jain A.K., Prabhakar S., Hong L., Pankanti S., FingerCode: a filterbank for fingerprint representation and matching, *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on Volume 2*, (1999).
4. Blum A., Langley P.: Selection of relevant features and examples in machine learning: *Artificial Intelligence*, 97(1-2), (1997) 245-271.
5. Kim K., Han I.: Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index: *Expert Systems with Applications*, 19(2), (2000) 125-132.

6. Liu H., Motoda H.: Feature transformation and subset selection: *IEEE Intelligent Systems and Their Applications*, 13(2), (1998) 26–28.
7. Hsieh C.-T., Lai E., Wang Y.-C., An effective algorithm for fingerprint image enhancement based on wavelet transform, *Pattern Recognition*, vol.36, (2003) 303-312.
8. M. N. Do, Directional multiresolution image representations, PhD thesis, EPFL, Lausanne, Switzerland, (2001).
9. Haykin S., *Neural Networks: A Comprehensive Foundation.*, New York: Macmillan, (1994).
10. Charalambous, C.; Conjugate gradient algorithm for efficient training of artificial neural networks; *Circuits, Devices and Systems*, IEE Proceedings G; vol. 139(3), (1992) 301–310.
11. Jain A.K., Prabhakar S., Hong L., Pankanti S., Filterbank-based Fingerprint Matching, *IEEE Transactions on Image Processing*, vol. 9(5), (2000) 846-859
12. Salil Prabhakar. Fingerprint Classification and Matching Using a Filterbank, Michigan State University, Ph.D. thesis, (2001).
13. Altun A.A., Allahverdi N., Recognition of Fingerprints Enhanced by Contourlet Transform with Artificial Neural Networks, 28<sup>th</sup> International Conference on Information Technology Interfaces, (2006) 167–172.

# Why Automatic Understanding?

Ryszard Tadeusiewicz and Marek R. Ogiela

AGH University of Science and Technology,  
Institute of Automatics, Al. Mickiewicza 30, PL-30-059 Krakow, Poland  
{rtad, mogiela}@agh.edu.pl

**Abstract.** In the paper a new way of intelligent medical pattern analysis directed for automatic semantic categorization and merit content understanding will be presented. Such an understanding will be based on the linguistic mechanisms of pattern interpretation and categorisation and is aimed at facilitation of in-depth analysis of the meaning for some classes of medical patterns, especially in the form of planar images or spatial reconstructions of selected organs. The approach presented in this paper will show the great possibilities of automatic lesion detection in the analysed structures using the grammar approach to the interpretation and classification tasks, based on cognitive resonance processes. Cognitive methods imitate the psychological and neurophysiological processes of understanding the analysed patterns or cases, as they take place in the brain of a qualified professional.

## 1 Introduction

Typical applications of Artificial Intelligence (AI) methods in biomedicine (e.g. medical diagnostics), in the area of engineering problems (e.g. computer systems security) and also in intelligent economic information systems include some traditional techniques: intelligent data processing and analysis, pattern recognition, neural networks, genetic algorithms and expert systems. Data processing and analysis provide us with better description of the objects or processes under consideration. Pattern recognition facilitates its classification – for example in automatic diagnostics. Neural network helps us to build behavioral models for control or forecasting. Genetic algorithms can solve optimization problems. Expert systems can advise us, what we ought to do in particular situations. This short outlook presents general view over a typical AI landscape.

In many biomedical, economical, and engineering problems these traditional techniques are completely sufficient. If we can build AI tools for intelligent data analysis, recognition and modeling, we are happy. This is true, but definitely not in all situations. Sometimes solving of complex problems leads to the necessity of **understanding** some signals, patterns and situations instead of simple processing, classification and interpretation. In fact problem understanding is the first necessary step for intelligent solving of the problem, when we use natural (not artificial) intelligence. Problem understanding is something more than signal processing – it needs also some knowledge and it demands special type of data processing. Details of natural understanding

are very complicated and obscure. Therefore, we can talk about understanding in terms of psychology and in frames of cognitive science, although in fact we can not understand the natural understanding process!

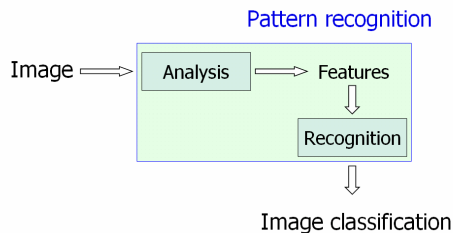
Nevertheless, we can propose efficient methods of artificial imitation of understanding. Although it sounds strange - in fact it is definitely possible to build the system for automatic understanding of selected data, signals and situations. This fact was proven by the authors in many previous papers and books on the basis of many examples of medical images. If computer powered by special AI programs can understand the nature of disease on the basis of the analysis of features of some medical images it can also be used to solve other complex problems, demanding automatic understanding. In the paper we describe the general methodology of the automatic understanding and we show how to use this methodology for solving selected biomedical, economical and also engineering problems.

## 2 The Idea of Automatic Understanding

Trying to explain what automatic understanding (AU) is and how we can force the computer to understand the image content we must demonstrate the fundamental difference between a formal description of an image and the content meaning of the image, which can be discovered by an intelligent entity, capable of understanding the profound sense of the image in question. The fundamental features of automatic image understanding can be listed as follows:

- We try to imitate the natural way in which a qualified professional thinks.
- We make a linguistic description of the image content, using a special kind of an image description language. Owing to this idea we can describe every image without specifying any limited number of *a priori* described classes.
- The linguistic description of an image content constructed in this manner constitutes the basis for the understanding of image merit content.

### Classical Pattern Recognition Process



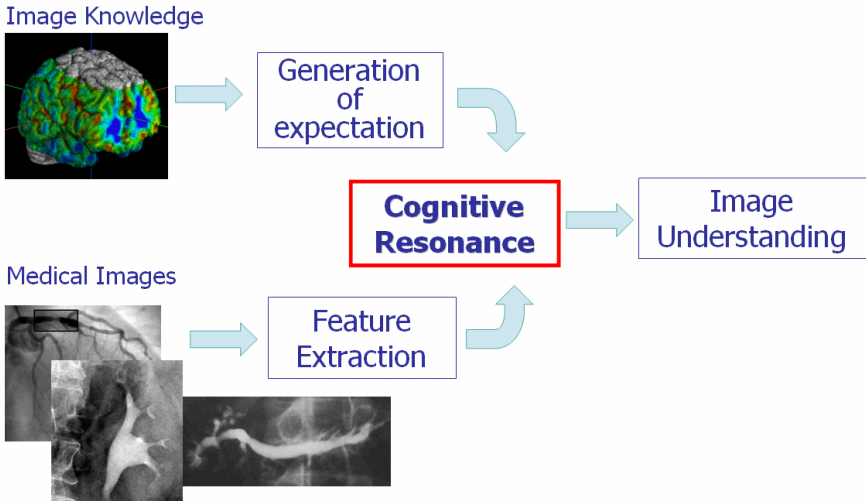
**Fig. 1.** Traditional method of medical image recognition

The most important difference between all traditional methods of automatic image processing and the new paradigm for image understanding is that there is one directional scheme of the data flow in the traditional methods while in the new paradigm

there are two-directional interactions between signals (features) extracted from the image analysis and expectations resulting from the knowledge of image content, as given by experts. In Fig. 1 we can see a traditional chart representing image processing for recognition purposes.

Unlike in this simple scheme representing classical recognition, in the course of image understanding we always have a two-directional flow of information (Fig. 2).

## Two-way Flow of Information in the Image Understanding Process



**Fig. 2.** The main paradigm of image understanding

In both figures we can see that when we use the traditional pattern recognition paradigm, all processes of image analysis are based on a feed-forward scheme (one-directional flow of signals). On the contrary, when we apply automatic understanding of the image, the total input data stream (all features obtained as a result of an analysis of the image under consideration) must be compared with the stream of **demands** generated by a dedicated **source of knowledge**. The demands are always connected with a specific (selected) hypothesis of the image content semantic interpretation. As a result, we can emphasise that the proposed 'demands' are a kind of postulates, describing (basing on the knowledge about the image contents) the desired values of some (selected) features of the image. The selected parameters of the image under consideration must have desired values when some assumption about semantic interpretation of the image content is to be validated as true. The fact that the parameters of the input image are different **can** be interpreted as a **partial** falsification of one of possible hypotheses about the meaning of the image content, however, it still cannot be considered the final solution.



Due to this specific model of inference we name our mechanism the ‘cognitive resonance’. This name is appropriate for our ideas because during the comparison process of the features calculated for the input image and the demands generated by the source of knowledge we can observe an amplification of some hypotheses (about the meaning of the image content) while other (competitive) hypotheses weaken. It is very similar to the interferential image formed during a mutual activity of two wave sources: at some points in space waves can add to one another, in other points there are in opposite phases and the final result is near zero.

Such a structure of the system for image understanding corresponds to one of the very well known models of the natural human visual perception, referred to as ‘knowledge based perception’. The human eye cannot recognise an object if the brain has no template for it. This holds true even when the brain knows the object, but shown in another view, which means that other signals are coming to the visual cortex. Indeed, natural perception is not just the processing of visual signals received by eyes. It is mainly a mental cognitive process, based on hypotheses generation and its real-time verification. The verification is performed by comparing permanently the selected features of an image with expectations taken from earlier visual experience.

Our method of image understanding is based on the same processes with a difference that it is performed by computers.

### **3 Linguistic Description of the Image in Understanding Procedures**

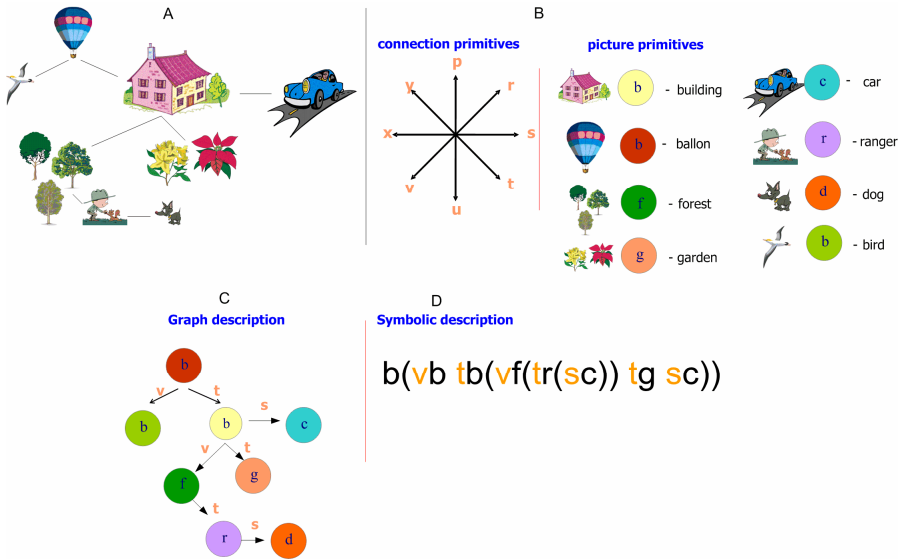
The close connection between our whole methodology and mathematical linguistics, especially a linguistic description of images, is a very important aspect of the automatic image understanding method. There are two important reasons for the selection of linguistic methods of image description as a fundamental tool for understanding images.

The first one results from the fact, that during the understanding process no classes or templates are known *a priori*. Indeed, the possible number of potential classes tends to infinity. So we must use a tool that offers us the possibilities to describe a potentially infinite number of categories.

The second reason owns to the fact that in the linguistic approach, after processing, we obtain a description of the image content without the use of any classification known *a priori* due to the fact that even the criteria of the classification are constructed and developed during the automatic reasoning process. This is possible due to a very strong generalisation mechanism involved in the grammar parsing process.

The only problem consists in a correct adjustment of the terms and methods of formal grammars and artificial languages when applying them in the field of images. The problem is very well known to specialists but for the completeness of our presentation let us explain some fundamental ideas.

When we try to build a language for the image description we must start with fundamental definitions of elements belonging to the suitable graph grammar. Let us assume that we must build a grammar for the description of a class of a landscape, similar to the image presented in Fig. 3.



**Fig. 3.** Initial stages used for syntactic description of the images. A) Analyzed scene, B) Elements of used vocabulary (picture primitives stands for “nouns” and connection primitives stands for “verbs” of the exemplary graph-grammar), C) Symbolic representation of the scene before describing it in terms of graph-grammar, D) Conversion of a graph diagram of the image into its symbolic description (string to be parsed for automatic understanding of the image merit content).

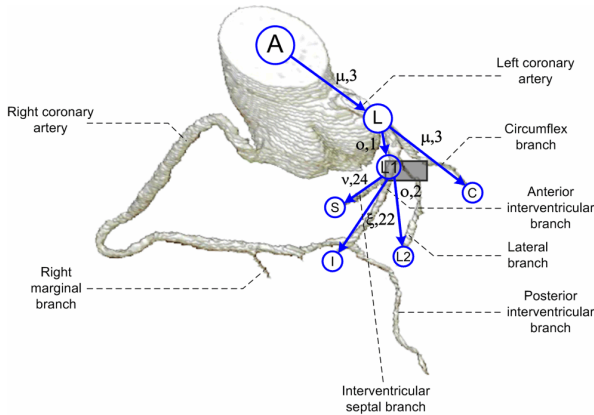
The analysis of the scene under consideration shows that we have some classes of graphic objects (‘primitives’) which can be built into the grammar as substantives (nouns). We also have some classes of relations between objects, which can be treated as the verbs of our grammar. So the vocabulary of grammar for the images under consideration can be shown as in Fig. 3.B. Using the proposed vocabulary we can replace landscape image with an equivalent scheme for the grammar, as shown in Fig. 3.C. On the basis of such a symbolic description we can also use the symbolic notations for elements of vocabulary; for every image we obtain a representation in terms of terminal symbols belonging to the definition of the grammar used (see Fig. 3.D).

After a final description of the image, using elements of a selected (or built for this purpose) image description language, we must implement the cognitive resonance concept. During cognitive resonance we must generate a hypothesis about semantic meaning of the image and we must have an effective algorithm for its on-line verification. Both mentioned activities are performed by the parser of the grammar used. Hypothesis generation is related to the use of a selected production (mappings included into formal description of the grammar). Verification of the hypothesis is performed by the incessant comparing of the selected features of the image to the expectations taken from the source of knowledge (e.g. qualified professionals).

## 4 Understanding Medical Visualization

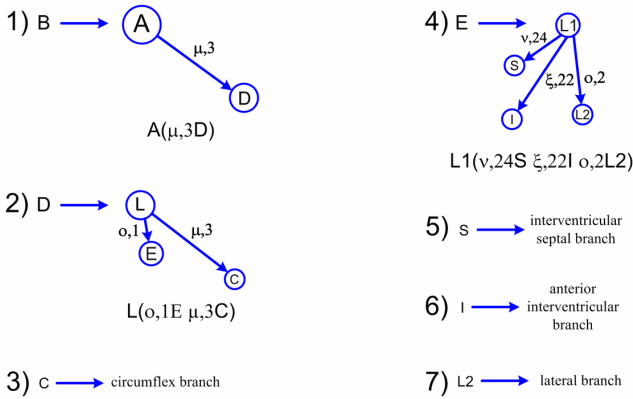
The details of automatic understanding of the medical images (in particular medical diagnostics imaginations) are explained in papers listed in the bibliography of this paper and also on the author page: <http://www.agh.edu.pl/uczelnia/tad/> , but the main idea can be shortly presented as follows. AU is based on the “cognitive resonance” process (described below), performed when the input stream of the data representing selected features encountered on input image is interfered with second stream of information, starting from the special form of grammatical representation of the expert knowledge (see fig. 2). Expert knowledge deposited in the computer program in grammatical form help us understood the merit content of the image during parsing of the input data stream presented for this purpose in special linguistic form. This two-stream based approach should be compared to traditional pattern recognition way, which is base on successive processing of one stream of information only, which is recollected on fig. 1.

Automatic understanding approach may be applied to medical visualization analysis because such images have a deep semantic meaning. The Authors have a great experience in application of such a way of semantic analysis for interpretation of medical images [5, 6, 7]. Below, an example of understanding of spatial coronary vessels reconstructions will be presented.



**Fig. 4.** Spatial labelling of left coronary artery and relations occurring between them. The grey rectangle marks the place of a major stenosis in a coronary artery.

For the coronary vessels analysis the proper graph-based grammar describing these structures was defined in such a way that the individual branches of the graph in the description identify all start and end points of coronary vessels and all bifurcations or transitions of main vessels into lower-level ones [3]. Thus developed graph-based structure will constitute the elements of the language for defining spatial topology and



**Fig. 5.** The set of productions in the form of graphs deriving the structure of coronary vessels

correct vascularization of the heart muscle together with the potential morphological changes, e.g. in the form of stenoses in their lumen (Fig. 4).

To define the location where the vessel pathology is present in the graph of coronary arteries the following grammar is proposed.

$$G_{edNLC} = (\Sigma, \Delta, \Gamma, P, Z), \text{ where}$$

$\Sigma = \Sigma_N \cup \Sigma_T$  and is a set of both terminal and non-terminal node labels describing the examined graphs and defined as follows.

$\Delta = \Sigma_T = \{\text{left coronary artery, anterior interventricular branch, circumflex branch, lateral branch, interventricular septal branch}\}$  – a set of terminal node labels.

$\Sigma_N = \{A, B, C, D, E, I, L, L1, L2, S\}$  is a set of non-terminal node labels.

$\Gamma = \{\mu,3; v,24; \xi,22; o,1; o,2\}$  is a set of labels describing edges of the graph.  $Z = \{B\}$  is the original starting graph.  $P$  – is a finite set of productions recorded in the graph-based form and presented in Fig. 5.

When we have full linguistic description of the medical object under consideration we can start next step leading to automatic understanding of the merit content of the image: the parsing process. This process is very similar to that one, which is performed by the compilers of most programming language. It also can be performed by the same software tools, which are used for compilation of source program to the binary codes. It is only one presumption which must be satisfied by the tool used for our purpose: The parser must be grammar-driven with the easy method of the exchange of grammar productions according to the different grammar description of different classes of analyzed images. For example one of most useful tools in our research was YACC.

During the parsing process driven by special graph grammar (which include knowledge of expert, e.g. professional radiologists) we try to perform translation between linguistic description of the image **form** (based on the grammar described above) toward the semantic description depicting **merit (medical) content** of the image, based on the experts knowledge.

The details of this process are rather complicated because of complicated nature of the used experts knowledge, nevertheless the general idea is very similar to the typical translation from one language to the another. During typical translation process we have the source text given in one language (natural or artificial) and we try to obtain the same sense expressed in another language (also natural or artificial). During the automatic understanding process we have also as a start point some text (linguistic description of the image under consideration) and we are going to obtain the same sense described in the language based on terms including elements of intelligent description of the image merit content (e.g. medical diagnosis based on the image).

The parsing process is named “cognitive resonance” because during this process we do make in fact some kind of “interference” between stream of structural data taken from the input image and another stream of semantic hypothesis and generated structural expectations, which is originated by the internal source of knowledge. For some merit hypothesis (connected with particular merit interpretations of the image content) we can observe the reciprocal elimination process, when the elements of structural description of the actual input image do not convolve with the expectations taken from the source of knowledge. The success we can obtain, when in at least for one semantic hypothesis the general (knowledge based) expectation agree with most of the elements of structural description of the image, which can be observed as an “resonance peak”.

## 5 Generalization of the Proposed Methodology

Basing on the general pattern understanding concept we have proposed several methods of automatic understanding of many kinds of information. We have introduced a general class of Understanding Based Decision Support Systems (UBDSS) which include Understanding Based Image Analysis Systems (UBIAS) described above and also another UBDSS type systems, listed in fig. 6.

We can not describe all above mentioned systems because of limited space of this paper. Nevertheless we want to pay special attention to the new class of intelligent Information Systems for economic purposes i.e. UBMSS (Understanding Based Managing Support Systems). Such systems are also based on cognitive resonance, but instead of medical images and illnesses important for UBIAS, UBMSS are dedicated to describe the states of business processes, which require understanding and interpretation, especially in case of strategic decision supporting [8].

The difference between a traditional DSS (Decision Support System) for economic purposes and the UBMSS proposed by us becomes visible when the computer on its own, without the human participation, attempts to describe the properties and consequences of the ratios computed. The results of automatic interpretation are expressed in the categories of the applied description language for the interpreted data properties.

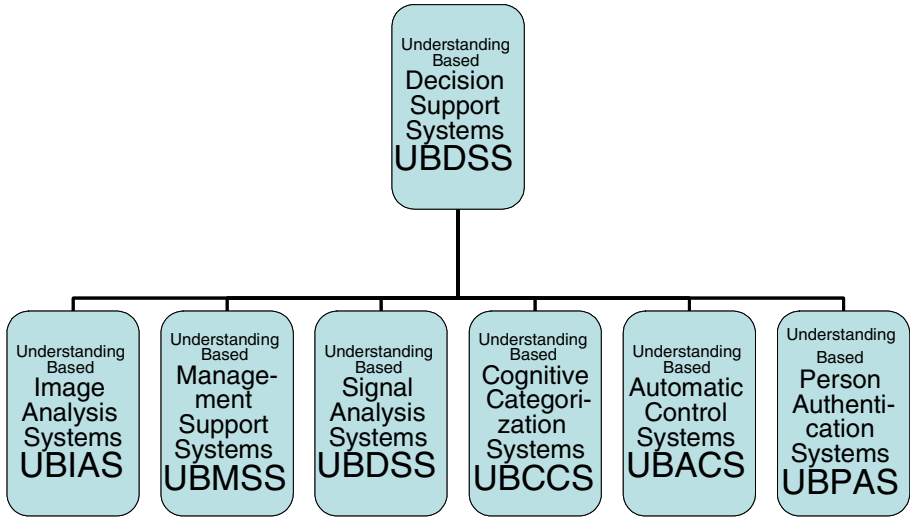


Fig. 6. General structure of set of UBDSS-class systems

## 6 More Detail Description of the UBMSS Idea as an Example of Using Methodology Under Consideration to the Economical Problems

We shall now present a proposed structure and operational methods of the already introduced AU-type systems for economical purposes. First, in order to systematise our considerations and to establish a reference point, let us recall a traditional (nowadays applied in practice) structure of economic information system application: computers are, obviously, involved since they are the ones that store and process data as well as analyse data in various ways. Information obtained from such computer systems is entirely sufficient for an effective management of business processes at the tactical level (as marked on fig. 7).

On the other hand, if we talk about management at the strategic level, we find out that despite automated data collection, storage and analysis, the task of business meaning understanding of the said data is in traditional systems the unique area of people (experts). So is taking and implementation of strategic decisions: this belongs only to people holding appropriate, high positions. The structure of such traditional IS, as presented on fig. 7 will be the starting point to propose a general structure of an AU-type information system.

In proposed AU based decision support system (UBMSS), whose structure has been presented on fig. 8, the initial processes of storing and pre-processing phenomena taking place in the analysed business entity, are analogous to the one we are dealing with in the traditional systems. The only difference is that with the perspective of automatic interpretation of data analysis process results, one can compute and collect a larger number of ratios and parameters since the interpreting automaton will not be dazzled or perplexed by an excess of information. This is what happens when people,

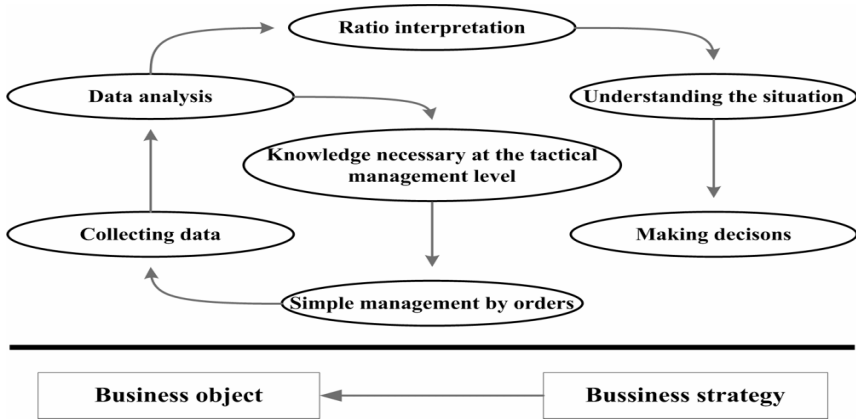


Fig. 7. Division of functions between people and computers in a traditional IS supporting business decision-making

interpreting situations, are ‘bombed’ with hundreds of ratios among which they can hardly find the important ones and then need to make a huge effort in order to interpret them correctly. There may also be no change to the business process management at the tactical level. This was left out of fig. 8 entirely since the AU information systems concept does not refer to this level at all.

The difference between a traditional system and the AU system becomes visible when the computer on itself, without human participation, attempts to describe the properties and consequences of the ratios computed. The results of automatic interpretation are expressed in the categories of the applied description language for the interpreted data properties. The above-mentioned language is a key element at this stage. It must be designed with great know-how. Its construction must therefore be based on collecting and systematising expert knowledge. Referring to the analogy with medical image automatic understanding systems, which were earlier said to be the area of some fully successful implementations of ideas described here, one could say that just like in medical systems [7], the basis for the development of the language subsequently used for semantic image interpretation (and diagnosing a disease) were some specified **changes** in the shape of the analysed organs. In the AU-type information systems the basic constructing units of the developed language should be **changes** of some specified business indicators.

Of course, focusing attention on a business index and ratio changes computed in the input part of the analysed AU-type information system, as only on those elements, which should be the basic components of an artificial language, is just the first step. The listing and appropriate categorisation of changes that should be registered in linguistic business processes corresponded only to the stage at which one defines the alphabet to be later used to build words and sentences, i.e. the language main object. In order to make it possible to create from the elements of this alphabet counterparts of words and sentences for subsequent use by the AU information system to describe the states of business process, which require understanding and interpretation, an introduction of additional mechanisms is needed. These mechanisms would enable combining the above-mentioned sentences into larger units. Therefore, at a level superior

to the above-described alphabet one must build the whole grammar of rules and transformations. This grammar can be used to create complete languages of description expressing important content, necessary to understand automatically the analysed processes.

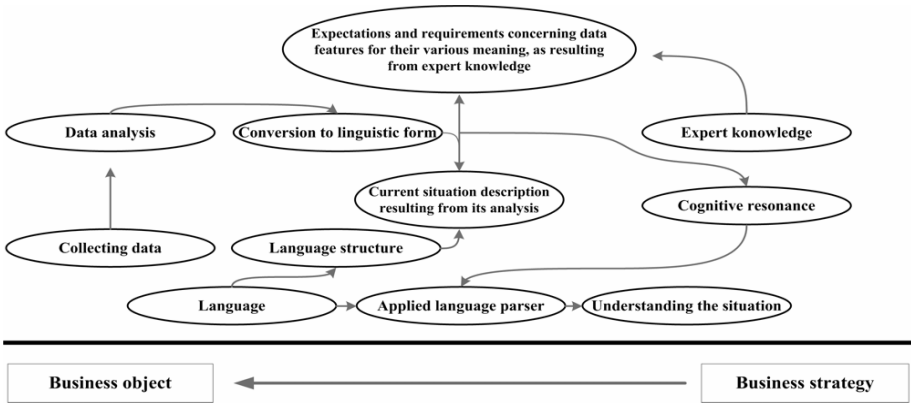


Fig. 8. General AU-type economic information system structure

In medical image understanding systems we constantly refer to, at our disposal were tools detecting local changes in the shape of some specified internal organs and their morphologic structures [5,7]. To understand the state of a given organ correctly, one needed to add to these graphic primitives their mutual spatial relations and combine them with anatomy elements. Owing to a definition of rules and the grammar constructs connected with them, one could combine for example a graphic category ‘change of edge line direction of a specified contour’ with a meaning category ‘artery stenosis anticipating a heart failure’.

Similarly, by building into the proposed language grammar the ability to associate business changes detected in various parts of the managed company and its environment as well as the possibility to trace and interpret time sequences of these changes and their correlations, it will be possible, for example, to understand what are the real reasons behind poorer sales of goods or services offered. As a result it will be, for example, possible to find out about the fact that this is due to the wrong human resources policy rather than the wrong remuneration (bonus) system.

After the development of an appropriate language which will (automatically!) express semantically oriented descriptions of phenomena and business processes detected in the business unit (e.g. a company) as supervised by the information system, a further AU information system operation will be very similar to the structure in which function the medical systems previously built by this system authors, as described in the previous section. The starting point for the business data automatic interpretation process, the process finishing with understanding their business meaning, is the description of the current state proposed in the system. It is expressed as a sentence in this artificial language, built specially for this purpose. Without going into details (described, among others, in earlier publications listed in the bibliography) one



can say that the above-mentioned language description for a human being is completely illegible and utterly useless. A typical form of such notation is composed of a chain of automatically generated terminal symbols. Their meaning is well based in the mathematically expressed grammar of the language used. Yet from the human point of view this notation is completely illegible.

A condition to mine the meanings contained in this notation and to present them in a form useful, by giving the necessary knowledge necessary to develop a new strategy concept, to people (those who take decisions at appropriate levels), is to translate these symbolic notation into a notation understandable for people. For this purpose two elements, shown in fig. 8, are necessary.

The first of these elements is duly represented knowledge of people (experts) who based on their theoretical knowledge and based on practical experience could supply a number of rules. Those rules state that in some circumstances, whose meaning interpretation could be described in detail, some particular features and properties should be found in the input data; those would be described with the use of a selected language. Now we have a description of the real situation, generated by tools founding their work on the results of business data analysis; it is generated with the use of the language we developed. We also have a set of hypothetical situations that carry some specified meaning connotations, which came into existence owing to the use of expert knowledge. We can therefore check in which areas these two descriptions converge (that increases the credibility of some semantic hypotheses [5, 6]) and in which the descriptions are contradictory. The latter case forms grounds to exclude other hypotheses and to narrow down the field of possible meanings.

The process of mutual interference between input information stream and the stream of expectations generated by an external knowledge source of the system results in the development of 'resonance peaks' in these areas in which the real situation 'concorde' with some specified expectations. These expectations ensue from the knowledge gathered before; in earlier works they have been called the cognitive resonance. Cognitive resonance happens in the course of the iteration process of comparison between features computed for input data and features theoretically forecast. Nevertheless we can also expect a possibility that the process will not always be convergent and the result will not always be unique. Yet in most practical implementations researched by experiments, the authors have managed to obtain the desired convergence and cognitive resonance uniqueness. As a result it mined out from the input data (in most research the data was medical images) information necessary to give the data correct interpretation in the interpreted meanings area, that is to lead to a situation in which the system understands the data and that it will be able to suggest to people the correct semantic interpretation.

The second AU information system element specified on Fig. 8 whose meaning has to be explained is a parser translating internal description languages into a form understandable for a human being. The parser concept, treated as a translation automaton steered by the used grammar syntax, translating some language formulas from an encoded into an executable form, has been discussed in Sect. 4.

In the AU information system described here the role of the parser is greater since its operations are steered to a significant extent by the cognitive resonance mechanism. In fact, in the cognitive system, the parser performs primarily the structural and meaning entry analysis. These entries were automatically generated in a special

artificial language noting important semantic facts. Nevertheless the AU system (UBMSS type) parser performs the above-mentioned meaning analysis as if as a side action since its basic role in the described diagram. As a translator, it receives an abstract code as its input. The code describes, in a language developed especially for this purpose, the current business situation. The output is to be the meaning of this situation specified in manner useful for men. The need for this meaning conversion from one language into another results from the fact that an artificial language developed to generate internal descriptions of the analysed business phenomena is constructed to obtain uniquely and effectively (automatically!) symbolic entries registering all important business process properties. These are obtained on the basis of the analysed data. This kind of meaning code can be built but essentially it is not understandable for people. Were its form understandable, it would not be very effective in the course of internal analysis leading to the cognitive resonance outcome.

Fortunately, during the translation there is a confrontation between the current description of the analysed business situation and the model entries resulting from expert knowledge. As a result of that we obtain the above-mentioned cognitive resonance but also entries generated automatically in this artificial, not understandable language are converted into entries legible for a human being. Their interpretation is now understandable. Based on these entries, the outcome of the parser's operation, one obtains the necessary knowledge. Subsequently, when one already understands what is taking place in the information systems (IS), one can make strategic decisions. No one would dare to transfer the very last step to the machine. This is among others because there is a need to take responsibility for the decisions taken and it would be hard to sue computer software.

## **7 Some Remarks About the Application of UBDSS Systems in the Technology - Information System Security Domain**

General idea of the UBDSS systems can be adapted to many technological applications. One of them definitely can be computer security problem. In this paper we show only few remarks about this on the base of selected example (intelligent spam filtering problem), but in further papers we will extend this idea for very general computer security problems.

Let us start from some general remarks. In fighting computer contamination, AU is not the just another way. It is not even the way but the only way to go. It is the way we always traveled fighting intuitively computer viruses, Trojan horses, worms, etc. We needed to understand them automatically without much of human intervention. What we propose here is a more systematic and conscious approach based on an emerging method which is considerable achievements in medical imagery, mentioned in Sect. 5.

There is very little doubt that we are very fast approaching or even most likely passed a point of time when our personal computers are busier fighting malware than conducting any useful computations. One may even wonder if computers are so good in defending themselves why to even bother to do anything about it. Unfortunately, it is not only our computer time which is wasted but our own as well. Each of us needs to spend probably close to one hour per day for verifying whether or not our defence

actually works. Our wasted time is ever increasing since illegal attacks become more and more sophisticated. Unless something drastic is done soon, we may face a real disaster since massive computer contamination can become a dangerous weapon in the hands of terrorists.

In this fight we do need new weapons, because traditional ones are insufficient. Moreover the most recent shift of spam from text to graphics shouts for such actions. Fortunately described here automatic understanding (AU) of images technology, recently developed by the authors and described in many papers, can be very helpful. Until now the most successful application of AU was shown for medical images, described above, and in economical decision support systems also mentioned in previous section. In case of medical images it is important to diagnose a malignant tumor on the base of understanding of the nature of pathological processes resulting particular form of medical image of ill organ, instead of trying to recognize the shape of the organ on base of some expected templates or learned patterns – because the form and localization of the cancer is unpredictable. Similar situation is in automatic understanding dedicated to strategic economical decision support systems.

The methods used by the authors for automatic understanding of medical images and proved as useful tool for many medical problems, can be used also for defending our computer against new forms of spam, containing not wish information in form of graphics (instead of text) and therefore not recognizable by typical anti-spam software. The methodology of automatic understanding (AU) of the images is very similar to the methods used by intelligent viewer (for example experienced doctor) during the visual inspection of the image under consideration.

The same approach we do propose for the anti-spam software. Thanks to the linguistic form of representation of the input data it is possible to describe infinite (!) number of spam examples by means of finite number of grammatical rules. Moreover, the software based on described methods can made differentiation between spam and useful information on the base of understanding of the merit content of every email under consideration. It will be done in similar way as act an intelligent reader of the information. This approach is very different from actual automatic filtering methods, which are concentrated on the form of the information disregarding it merit sense. Surprisingly, the underlying medical philosophy: “do not make harm” is also applicable to spam.

We hope AU can help us to determine the meaning of the analyzed data of any format including images. It has roots in cognitive methods related to psychological and neurophysiological processes of understanding the analyzed data. AU complements both Unified Modelling Language (UML) and the Rational Unified Process (RUP). When we can use our methodology to the real understanding of the email merit content (both for email containing only the text and form email which content is represented in graphical form) we can not only perform more effective spam detection, but also we can propose many intelligent application for grouping of the information, for selecting them on the base of merit criteria and for its summarization.

Sir Winston Churchill said (about some stage of the Second World War): *It is not the end, it is even not the beginning of the end. It is perhaps the end of the beginning.* We can declare about our proposition described above: It is not the end, it is even not the beginning of the end. It is merely the end of the beginning. Unlike the Churchill's situation, the struggle with the computer contamination will never end. This is why we have no choice but to develop better methods for AU.

## 8 Conclusion

In the paper we present a new approach to semantic data analysis and pattern understanding based on cognitive analysis using the linguistic approach. We have proposed the general methodology of machine cognitive inference which allows extracting the semantic information from the analyzed patterns. The developed class of Understanding Based Decision Support Systems (UBDSS) was proved to be useful both for medical, economical and also technological purposes. Applying such a methodology we successfully attempted to develop an experimental implementation of the IT systems relevant to many decision support problems including (but not limited to) the UBIAS and UBMSS systems. Construction of these systems consisted in developing a lab version of the appropriate data acquisition system for data originating from the supervision of health services diagnostic processes (UBIAS) and of developing the necessary grammar and knowledge base. The created system was applied for interpretation tasks, i.e. processing data carrying meaning. The obtained usefulness of correct input data interpretation in the form of multi-dimensional vectors determining semantic information amounted to 90.5 %.

**Acknowledgements.** This work was supported by the AGH University of Science and Technology under Grant No. 10.10.120.39.

## References

1. Albus, J.S., Meystel, A.M.: Engineering of Mind: An Introduction to the Science of Intelligent Systems, Willey (2001)
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classifications, 2nd Edition, Willey (2001)
3. Khan, M.G.: Heart Disease Diagnosis and Therapy. Williams & Wilkins, Baltimore (1996)
4. Meyer-Baese, A.: Pattern Recognition in Medical Imaging. Elsevier (2003)
5. Ogiela, M.R., Tadeusiewicz, R.: Nonlinear Processing and Semantic Content Analysis in Medical Imaging - A Cognitive Approach. IEEE Transactions on Instrumentation and Measurement. 54(6) (2005) 2149-2155
6. Ogiela, M.R., Tadeusiewicz, R., Ogiela, L.: Image Languages in Intelligent Radiological Palm Diagnostics. Pattern Recognition. 39 (2006) 2157-2165
7. Tadeusiewicz, R., Ogiela, M.R.: Medical Image Understanding Technology. Springer, Berlin-Heidelberg (2004)
8. Tadeusiewicz R., Ogiela L., Ogiela M.R.: Cognitive Analysis Techniques in Business Planning and Decision Support Systems. LNAI. 4029 (2006) 1027-1039

# Automatic Target Recognition in SAR Images Based on a SVM Classification Scheme

Wolfgang Middelmann, Alfons Ebert, and Ulrich Thoennessen

FGAN-FOM Research Institute for Optronics and Pattern Recognition, Germany  
Middelmann@fom.fgan.de, Alfons.Ebert@fom.fgan.de

**Abstract.** The performance of classifiers is commonly evaluated by classification rate and false alarm rate (FAR). Many applications like traffic monitoring, surveillance and other security relevant tasks suffer from the problem balancing the performance criteria in an appropriate way. In this contribution, we propose a kernel classification scheme with high performance in discriminating classes and rejecting clutter objects. Especially, it determines a class membership assessment. The classification scheme consists of two kernel classification stages and a maximum decision module as combiner. For tests, we use targets taken from the MSTAR synthetic aperture radar (SAR) dataset and clutter objects extracted from SAR scenes by a screening process. The dependency on parameter variations is shown and receiver operator characteristic (ROC) curves are given. The results confirm the high classification performance at low FARs. The integration into an operational demonstration system is in progress.

## 1 Introduction

Automatic target recognition (ATR) is a widely studied task in remote sensing. It finds a variety of civil and military applications like traffic monitoring and control, surveillance, and other security relevant tasks. Target Classification is an important component of each ATR-System. Due to their robustness and their generalization properties kernel machines as support vector machines (SVM) [8, 2], offer a chance for solving this classification problem very accurately and efficiently. Unfortunately, kernel machines do not facilitate the rejection of clutter objects. Especially, a class membership assessment is not exploited.

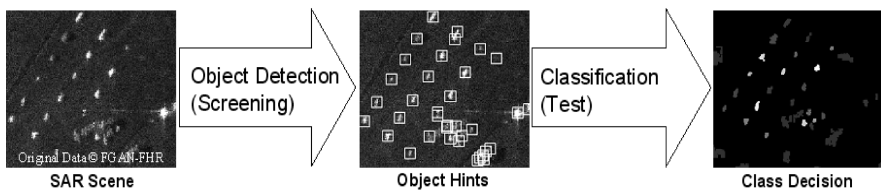
In this paper, we present a new technique that relies on both the class discrimination and the class membership. By taking more information about the classes' structure into account, this technique is able to provide high-performance ATR systems with low false alarm rates (FAR). This technique is realized as a multi stage classification scheme. The main advantages are the robustness and the straightforward adjustability to application demands.

In the following section, we present a target recognition problem and discuss application relevant requirements. An overview of common kernel classification approaches is given in section three. In section four, we introduce our new method.

Experimental results with targets and screened clutter objects are presented in section five. A resuming conclusion follows afterwards.

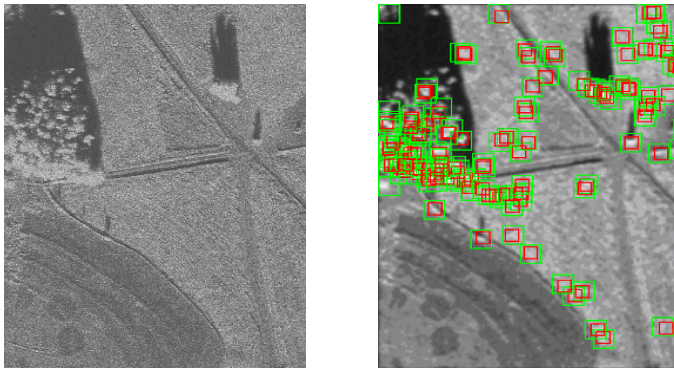
## 2 The Classification Problem in an ATR Framework

Target classification in synthetic aperture radar (SAR) scenarios is a main ATR task. In extended scenarios, regions of interest are detected containing target hypotheses that have to be classified. In the case of stationary targets screening algorithms, e.g. hot spot detection, can do this. Hence, the proposed processing chain in Fig. 1 consists of a screening process identifying target cues and a high-performance classifier.



**Fig. 1.** The target recognition processing chain consists of a screening and a classification module with integrated pre-processing

The screening module chooses the target hints. Fig. 2 (right) depicts an example. Green boxes in the smoothed SAR image mark the detected objects. The screening method should yield hints including all specified targets whereas the number of false object hints is of lower interest.



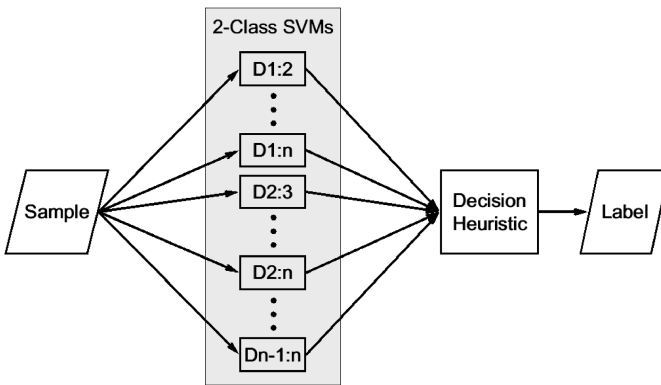
**Fig. 2.** Left: original SAR magnitude image, Right: Gaussian smoothed, screening hints (green boxes), gravity centered targets (red boxes)

Thus, our classification approach should have a high capability to reject clutter objects and it has to handle multi-class problems.

### 3 Existing SVM-Based Classification Approaches

Generally, the SVMs are only capable to discriminate between two classes. Solving a multi-class-problem a decision scheme is necessary. An overview of possible classification schemes is given in [4]. Three main categories are identified: parallel, cascading, and hierarchical. Common SVM schemes are the one-to-rest approach with maximum decision as combiner, the one-to-one method followed by a decision heuristic, and several hierarchical techniques like decision trees or hypercube schemes, see [8, 3]. Hierarchical classification schemes are not discussed in this paper, because they often suffer from single false decisions or have a computational expensive overlap like the hypercube model.

The one-to-rest approach, tests each class against the union of all other classes. Then, a maximum decision constitutes the final classification result. A rejection occurs, if the assessment of the accepted class is beneath a user-defined threshold. A drawback is that the outer bounds of the classes are approximated too rough. Only in a closed world like digit recognition, this is negligible. However, this method does not support the increment of further classes. Moreover, the computational complexity is rather high, as the training of each classifier is executed using the complete set of training samples.



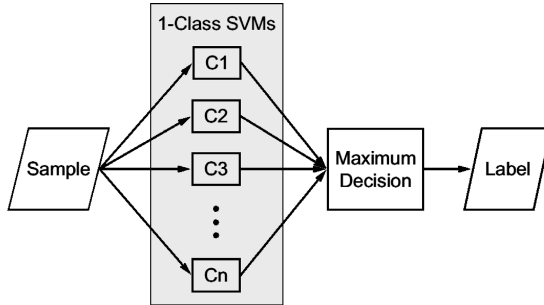
**Fig. 3.** One-to-one classification scheme with multi-class decision heuristic

The one-to-one approach, displayed in Fig. 3, tests each pair of classes. A voting scheme then determines the final classification result [5]. A rejection takes place either if the class selection is ambiguous or if a threshold dependent acceptance criterion in the kernel-induced vector space fails. The addition of further classes only needs a training of the supplementary basis SVMs. The computational complexity is lower than the one of the previous approach. Unfortunately, a shortcoming is the insufficient approximation of outer class boundaries.

All mentioned techniques do not determine a membership assessment, thus the comparison to clutter objects is unconfident.

Membership assessment is with the well-known 1-class SVM [6, 7] possible. Fig. 4 depicts the 1-class SVM classification scheme. At first, each class is evaluated by its

membership function. Then the maximum decision combiner takes place. An advantage is that the informational linking of the classes occurs at the end of the process. Therefore, the user can easily upgrade the classifier system with respect to new classes. The computational effort is relatively low. A drawback is that the classifier is not supported by information from class discrimination. A prerequisite of this method is a balanced scale of all membership assessments. Otherwise, objects positioned near the borderlines of two classes may be classified wrong. Unfortunately, there is no guarantee that the 1-class SVMs are fulfilling this demand.



**Fig. 4.** 1-class SVM scheme for multi-class problems

In the closed world case, the one-to-one scheme is preferable [3]. Methods suitable for problems with clutter objects rely on a confident class membership.

## 4 The Novel SVM21 Classification Scheme

The classification approach should have a high capability to reject clutter objects and it has to handle multi-class problems. Thus, class discrimination and membership assessment have to be supported. In the following, we present our new classification scheme and give algorithm details.

### 4.1 Scheme

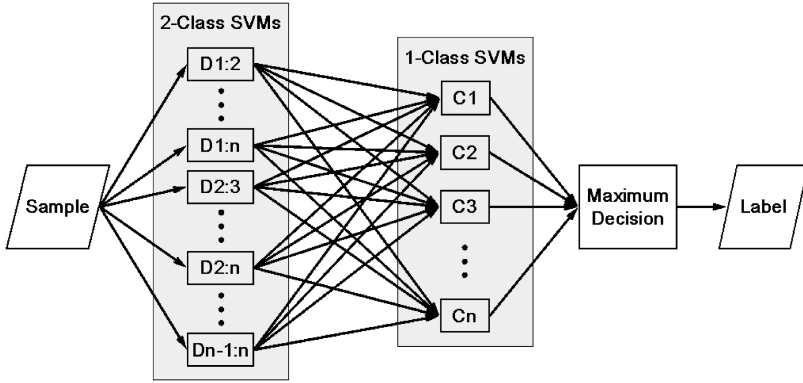
The class discrimination is realized by using 2-class SVMs in a pre-classification step. Class membership assessment is achieved by 1-class SVMs. Fig. 5 depicts the novel SVM21 scheme. A resuming maximum decision combiner determines the final classification and enables an easy to handle reject criterion.

### 4.2 Algorithm Details

Both, 1-class SVM and 2-class SVM training yields decision functions like

$$f(x) = \sum_{i=1}^l \alpha_i K(x_i, x) - b. \quad (1)$$





**Fig. 5.** Novel classification scheme with 2-class SVMs for class discriminating feature determination, 1-class SVMs for membership assessment, and a maximum decision combiner

Hereby the  $x_i$  are the original sample vectors,  $\alpha_i$  are Lagrange multipliers of the underlying dual optimization problem, and  $b$  is the so-called bias. Usually the kernel  $K$  depends on several parameters. In this investigation, we use the radial basis function (RBF) kernel  $K_\sigma(x,y)=\exp(-\|x-y\|^2/\sigma)$ , as RBF kernels should be utilized in particular if there is no other knowledge about the classes' structure, see Byun and Lee [1].

Then the classification of the original data  $o$  is achieved by performing two classification steps. At first, this original data  $o$  is pre-classified using a scheme of 2-class SVM decision functions for discriminating between all  $N = n(n-1)/2$  pairs of the  $n$  classes. By this, a new feature vector  $c(o)$  is generated with elements:

$$c(o) = (c_1(o), \dots, c_N(o))$$

$$c_i(o) = \sum_{j=1}^{n(i)} w_{ij} K_{\sigma_2}(o_{ij}, o) + w_{i0}, \quad i = 1, \dots, N \quad (2)$$

The samples  $o_{i1}$  to  $o_{i,n(i)}$  are those of the two classes defining the  $i$ -th pair of classes, and  $w_{i0}$  to  $w_{i,n(i)}$  are the weights of the appropriate 2-class SVM.

Based on this pre-classification result  $c(o)$  the final classification is accomplished. One possibility would be to use this complete feature vector. However, the separation of one class from all other classes is achievable using only the results of  $(n-1)$  classifiers. Especially, the outcomes of all other ones corrupt often the final decision. Therefore, class-dependant feature vectors  $c^{(k)}(o)$  are deduced from the full-size feature vectors  $c(o)$  consisting only of the results of the classifiers separating class  $k$  from all other classes. Afterwards, the 1-class SVMs determine membership assessments  $v_k(c^{(k)}(o))$  for each class  $k$  of the  $n$  classes operating on these new feature vectors  $c^{(k)}(o)$ :

$$v_k(c^{(k)}(o)) = \sum_{j=1}^{m(k)} u_j^{(k)} K_{\sigma_1}(c_j^{(k)}, c^{(k)}(o)) + u_0^{(k)}. \quad (3)$$

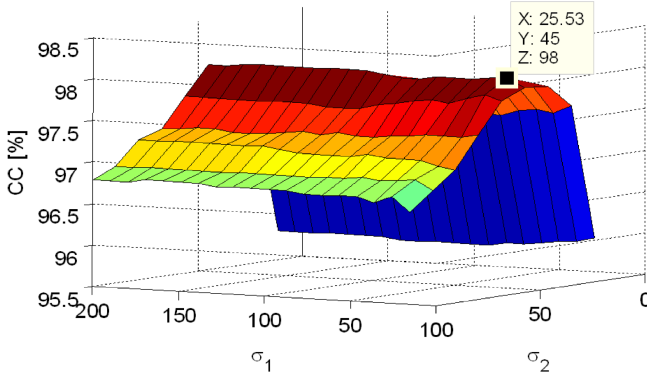
The  $c_1^{(k)}$  to  $c_{m^{(k)}}^{(k)}$  are those  $c^{(k)}(x_j)$  for all training samples  $x_j$  of the  $k$ -th class, and  $u_0^{(k)}$  to  $u_{m^{(k)}}^{(k)}$  are the weights of the appropriate 1-class SVM. Then the class  $k_{win}$  with the highest membership assessment  $v_k$  has to pass the reject criterion

$$v_{k_{win}}(c^{(k)}(o)) \geq TOL. \quad (4)$$

As the dimension of the deduced feature vectors  $c^{(k)}(o)$  is low, the computational effort is slightly higher than the one of our previous classification scheme. Finally, the novel SVM21 method depends on the kernel parameters of the 2-class SVMs and the 1-class SVMs, and the reject threshold TOL of the 1-class SVMs.

## 5 Experiments with SAR Data Inclusive Screened Clutter Objects

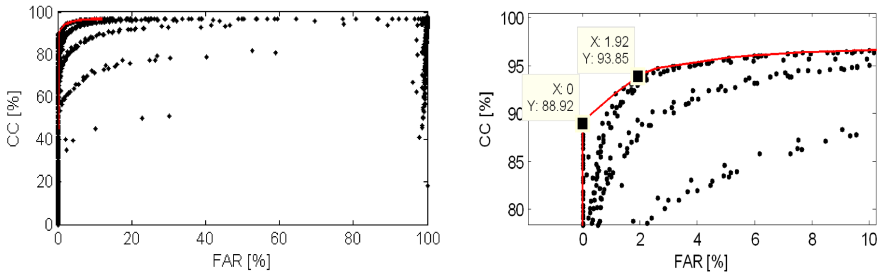
Experiments have been carried out with the MSTAR public target SAR dataset. We have chosen ten classes that is a lower limit in an operational environment. This dataset consists of 3671 training and 3203 test samples. The depression angle is  $17^\circ$  for the training data and  $15^\circ$  for the test data samples.



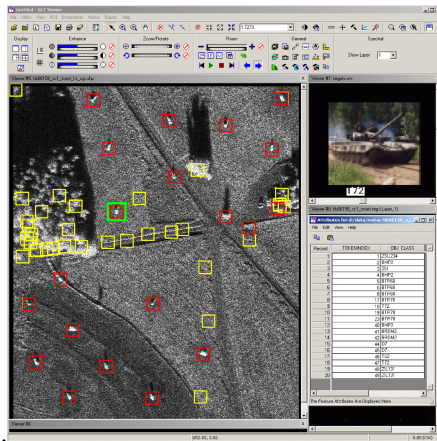
**Fig. 6.** Maximum CC (w.r.t. TOL) values of SVM21 for the MSTAR 10-class problem, visualization of the parameter dependence, closed world performance at 98.00% (CC is the z axis)

The closed world performance for the SVM21 is at 98.00%. The classification quality CC (correctly classified) depends on the three parameters  $\sigma_1$ ,  $\sigma_2$ , and TOL. In Fig. 6 CC is given by taking the maximums with respect to TOL for each  $\sigma_1$ ,  $\sigma_2$  parameter combination. The robustness of CC relative to  $\sigma_1$  variations is significant.

For investigations concerning low FAR, usually ROC curves are studied. Here the problem depends on three parameters, thus the naming ROC sets (or manifolds) is more adequate. The first MSTAR clutter scene set of 50 images has been used. From each scene, we have taken the 150 most significant confusion samples. They are centered and normalized like the target samples. Fig. 7 depicts the ROC set concerning the 10 target classes and the 7500 confusion samples. Additionally, an upper envelope is plotted in red for a better interpretation.



**Fig. 7.** ROC set of SVM21 for MSTAR 10-class problem with 7500 clutter samples taken into account, overview at the left side and details at low FARs at the right (FAR=x, CC=y axis)



**Fig. 8.** ERDAS user interface with classification results of the MSTAR 10-class problem, target vehicles in a clutter scene, accepted (red), rejected (yellow), and the selected (green) object

A high classification rate of about 94% was achieved at a FAR lower than 2%. In the special case of suppressing all false alarms, the classification rate only drops to circa 89%. In spite of the necessary data normalization, this is an acceptable result.

## 6 Conclusion

In this contribution, we present an ATR processing chain of screening and classification. It integrates the novel SVM21 approach for multi-class classification especially improving the rejection of clutter objects. An essential advantage is the usage of class discriminating features and the determination of a class membership assessment. By this, a maximum decision module is possible as combiner and an easy handling is given. The experiments have used targets taken from the MSTAR public SAR dataset and clutter objects extracted from MSTAR SAR scenes by the screening process. The ROC sets confirm the high classification performance at low false alarm

rates. Additionally, the classification quality is robust against variations of the parameter  $\sigma_1$ . Furthermore, the SVM21 consumes a moderate amount of computational effort.

In Fig. 8, an example of our ATR software demonstrator integrated in ERDAS is presented. The main window displays the SAR scene with the classification results. Yellow boxes mark rejected objects. Red boxes indicate accepted objects. The user has selected one object (green box). The upper right window visualizes the corresponding classification result. The lower right window shows further attributes.

Automatic parameter selection for our novel classification scheme is a future task. Other pre-classification concepts will be studied as well. The influence of further SAR parameters and SAR operation modes should be investigated. The integration into an operational demonstration system is continuing.

## References

1. H. Byun, S.-W. Lee, "A Survey on Pattern Recognition Applications of Support Vector Machines", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 17, No. 3, pp. 459-486, 2003
2. N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000
3. C. W. Hsu, C. J. Lin, "A comparison on methods for multi-class support vector machines", Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001
4. A. K. Jain, R. P. W. Duin, J. Mao, "Statistical Pattern Recognition: A Review", *IEEE PAMI*, Vol. 22, No. 1, pp. 4-37, 2000
5. W. Middelmann, A. Ebert, U. Thönnessen, "Kernel-Machine-Based Classification in Multi-Polarimetric SAR Data", *Proceedings of SPIE - Algorithms for Synthetic Aperture Radar Imagery XII*, E. G. Zelnio, F. D. Garber (eds.), 5808, pp. 247-256, 2005
6. B. Schölkopf, A. Smola, R. Williamson, and P. L. Bartlett, "New support vector algorithms" *Neural Computation*, 12, pp. 1207-1245, 2000
7. B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution", *Neural Computation*, 13, pp. 1443-1471, 2001
8. V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998

# Adaptive Mosaicing: Principle and Application to the Mosaicing of Large Image Data Sets

Conrad Bielski and Pierre Soille

Institute for Environment and Sustainability  
DG Joint Research Centre, European Commission,  
I-21020 Ispra (VA), Italy  
{Conrad.Bielski, Pierre.Soille}@jrc.it

**Abstract.** Automatic image compositing of very large data sets is necessary for the creation of extensive mosaics based on high spatial resolution remotely sensed imagery. A novel morphological image compositing algorithm has been developed which adapts to salient images edges. This technique produces seam lines that are difficult to identify by the naked eye which is also a characteristic to measure the quality of the resulting seam line. This paper begins with a description of the methodology and results based on Landsat 7 ETM+ imagery. It is also shown how updates to an already composited image data set can be easily made without having to reprocess the entire data set. Finally, ways of quantifying the quality of an automatically delineated cut line and future research directions are discussed.

## 1 Introduction

Image mosaicing (also called image compositing) is a technique that has been applied since the early beginnings of photography to join two or more images together. With photographs in-hand, it was necessary for the artist/photographer to cut the pictures in the overlapping region. The trick was to cut the overlapping photos so that the observer could not see that any such cut was made. This same technique was applied for mosaicing airphotos where a human operator was required to carefully define a seam line. This line crosses the overlapping domain by following a series of points corresponding to the same set of features in each image [1], [2]. Today, digital images still require seam lines to be defined in order to create mosaics but at the same time trying to leave out the human operator. Automatic seam line delineation is used in remote sensing applications as well as in the wider image processing domain. However, developing a satisfactory automatic seam line delineation algorithm to replace a primarily subjective human task depends on one's definition of 'best'. Milgram [3] defines the best seam line as minimizing the visual confusion created by the artificial edge along the seam. In other words, when a human looks at the joined images, it should be difficult to distinguish the location of the cut line. The cost of finding the 'best' seam line should also be taken into account even though attaching a price to a given algorithm and corresponding seam line is defined by the researchers

themselves. Reference [4] contains a short survey on image mosaicing while an up-to-date bibliography can be found in [5].

This paper focuses on a) the adaptive nature of the mathematical morphology based image compositing algorithm [4] for composing very large data sets of images and b) the ability to extend this algorithm to deal with the inclusion of new or updated imagery into an already composed data set. While the morphological gradient is currently used as the driver for the seam line delineation, potentially other image characteristics could also be used thus improving the algorithm based on the requirements of the user. This could be achieved by filtering the input imagery prior to compositing. Based on the 'best' seam line criterion, it will be possible to choose the most appropriate filter parameters.

The rest of the paper is organised as follows. In Sect. 2, a simplified image compositing algorithm is presented as well as the updating methodology when either new or alternate imagery is made available. The results based on the described methodology are presented in Sect. 3. They are followed by discussions on seam line delineation quality criteria and possible measures of seam quality (Sect. 4), and the conclusions are presented in Sect. 5.

## 2 Morphological Image Compositing

This section briefly describes a simplified version of the morphological image compositing technique [4] as well as its practical implementation.

### 2.1 Principle

Morphological image compositing produces adaptive seam lines in the sense that their positioning is driven by the image content. The underlying methodology is based on a region growing procedure initiated by seeds corresponding to those regions where there is no overlap. The growth is constrained by a gradient image so that it stops when growth fronts meet at high gradient values corresponding to object boundaries. Consequently, the seam line is barely visible in the resulting mosaic.

We now present a formal and simplified description of the methodology originally proposed in [4]. The method is iterative in the sense that the growth of the seeds operates in those regions where only two images overlap, then three, and so forth until the maximum degree of overlaps is reached.

- Input: arbitrary number  $n$  of input overlapping images denoted by  $f_1, \dots, f_n$ .
- We assume that none of the input images is fully included in the union of the other images. That is, in mathematical terms, for all  $i \in \{1, \dots, n\}$ ,  $\mathcal{D}_i \not\subseteq \bigcup_{j \mid j \neq i} \mathcal{D}_j$ , where  $\mathcal{D}_i$  denotes the definition domain of the  $i$ th image.
- Let  $\mathcal{D}_f$  denote the definition domain of the composed image:  $\mathcal{D}_f = \cup_i \mathcal{D}_i$ .
- Let us define a function  $g$  indicating how many images are available for each pixel  $\mathbf{x}$  of  $\mathcal{D}_f$ :  $g(\mathbf{x}) = \text{card}\{i \mid \mathbf{x} \in \mathcal{D}_i\}$ .
- The so-called mask image  $f_{\text{mask}}$  is used to constrain the growth of the markers (seeds) defined below. In its basic form, it is simply defined as the

pointwise minimum between the morphological gradients of all input images:

$$f_{\text{mask}} = \bigwedge_i \rho_B(f_i).$$

Since this mask image is at the root of the adaptive definition of the seam lines, it can be modified so as to maximise some function. This will be discussed in Sec. 4.

- We denote by  $f^{(k)}$  the values of  $f$  that are defined at the end of iteration  $k$ :  $\mathcal{D}_{f^{(k)}} = \{\mathbf{x} \mid g(\mathbf{x}) \leq k\}$ . Initially,  $k$  equals 1 since the values are only known where there is no overlap (i.e., those regions of the image  $g$  that have a value equal to 1).
- Let  $\mathcal{D}_i^{(k)}$  refer to those pixels of  $\mathcal{D}_{f^{(k)}}$  whose values originate from the input image  $f_i$ :  $\mathcal{D}_i^{(k)} = \{\mathbf{x} \in \mathcal{D}_{f^{(k)}} \mid f^{(k)}(\mathbf{x}) \leftarrow f_i(\mathbf{x})\}$ .
- The marker image at iteration  $k$  is then defined as follows:

$$f_{\text{marker}}^{(k)}(\mathbf{x}) = \begin{cases} i, & \text{if } \mathbf{x} \in \mathcal{D}_i^{(k-1)}, \\ 0, & \text{otherwise (i.e., no marker)}. \end{cases}$$

That is, initially, markers are defined only in those regions where there is no overlap. These initial markers correspond to the seeds initiating the region growing process.

- The following decision rule indicates the index  $i$  of the image that should be considered at a given pixel location  $\mathbf{x}$ :  $f^{(k)}(\mathbf{x}) = f_i(\mathbf{x})$ , the index  $i$  being defined by the label of the considered pixel in the image of grown regions:  $i = \left[ \text{CB}_{f_{\text{marker}}^{(k)}}(f_{\text{mask}}) \right](\mathbf{x})$ , where CB refers to the catchment basin known in the morphological watershed region growing procedure [6].

The method has also been adapted for the removal of specific objects such as clouds in satellite images [4].

## 2.2 Image Composition Implementation

**Input Imagery.** The automated image compositing procedure will be showcased on a subset of Landsat 7 Enhanced Thematic Mapper plus (ETM+) imagery. The original data set is made up of 724 scenes covering the entire territory of the EU.

**Required Preprocessing.** Some image preprocessing is required prior to delineating and cutting the input satellite imagery. Input images must be co-registered in a common reference coordinate system, identical spatial resolution, and same grid. Each image must have a Region Of Interest (ROI) created which is a binary image with foreground pixels being found where there is data in all 7 spectral bands of the ETM+ imagery. A cloud mask is generated automatically [7] (a binary image with pixels with clouds turned on) and then applied in the automatic image compositing procedure to minimise the appearance of clouds in the final mosaic.

**Actual Image Composition.** As proposed in [4], very large image sets cannot be processed entirely in computer memory (our test system has 4GB of RAM) and thus an order independent image compositing solution was devised [8]. The order independent implementation divided the compositing procedure into smaller pieces but at the same time created image composites that were equal to the original implementation. This solution was made possible by first generating an overlap matrix. The overlap matrix is a symmetrical  $n \times n$  indicator matrix indicating whether the definition domains of an arbitrary image pair overlap or not. This image overlap information drives the independent image compositing procedure and reduces the memory requirements to only the anchor image (the image for which cut lines are being delineated) and all images that overlap the anchor image. This permits the execution of the image compositing process in parallel which also reduces processing time.

Essentially, at each iteration of the image compositing procedure, the original anchor image ROI is updated as well as any overlapping image ROI's only within the definition domain of the anchor ROI. Cut line delineation begins in regions with the smallest number of overlaps and proceeds into regions ordered by the number of overlaps. This iterative procedure stops when the highest number of overlaps has been processed.

At the end of each iteration, an integrity check of the updated ROI's is made. Originally implemented for debugging purposes, it is now part of the processing chain and checks that neither the anchor image or any of its overlapping images have pixels which overlap along the cut line. The final output of the order independent image compositing procedure is a set of updated image ROI's which can be used to cut the original imagery to produce an image mosaic. This set of updated ROI's fit together perfectly along salient image features.

**Updating the Composited Image Database.** Updating of the composed image data set could be required when new imagery is made available. However, one does not want to reprocess the entire data set only to delineate the cut lines of the few additional images. At the same time, it is not possible to simply replace the original composed image with the new image based on the same updated ROI because the variability in image features may require new cut lines. Therefore, an updating procedure was necessary for the maintenance of the image mosaic. The updating procedure is also based on the overlap matrix. If the new image is simply a replacement (i.e., the new image location and dimensions are identical to the old one) then the same overlap matrix can be applied. Otherwise, an additional line/column must be added to the overlap matrix in order to take into account the newly added scene.

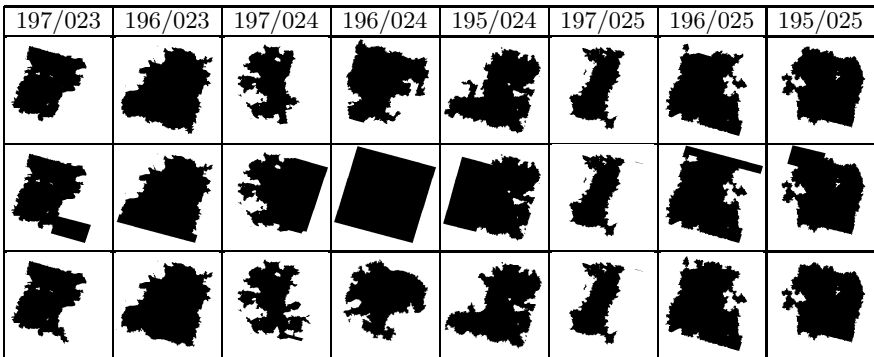
At first, the new image and associated overlap images must be prepared and entails updating all overlap image regions falling within the anchor image to their initial state. It is imperative that such updates are only made within the definition domain of the new image to be updated because the seam lines will only be computed within the anchor image (i.e., the new image to update the data set). Next, the update proceeds by running the order independent image compositing routine solely on the overlap matrix column (row) of the new anchor



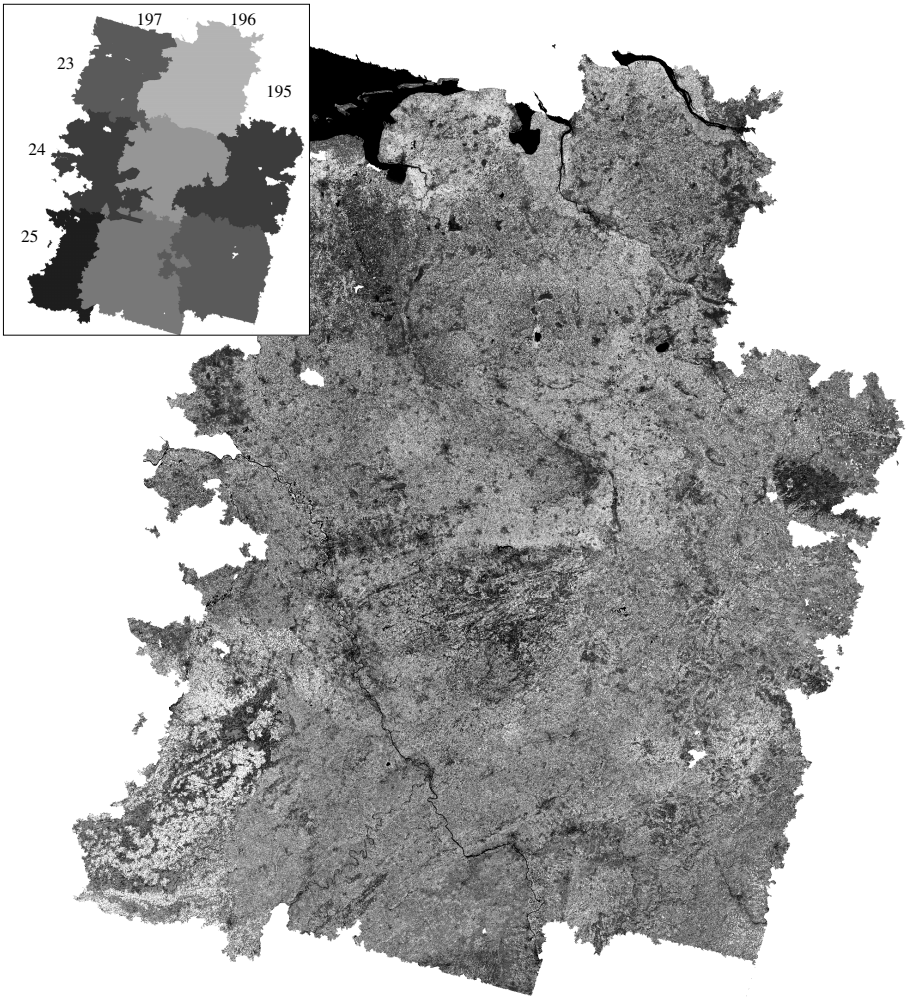
image. The seam line update processing can be distributed by blocks of rows (columns) which are ultimately stitched together.

### 3 Results

Figure 1 (top image row) presents the cut lines for the subset of images found within the Landsat ETM+ data set and are the results of applying the order independent image compositing procedure. After processing the data set however, it became known that one of the input images was not properly georeferenced prior to image composition. Consequently, a Landsat scene with the same path/row combination but different acquisition date was found and needed to be inserted into the data set. Prior to running the order independent image composition updating, the overlapping images had to be prepared (Fig. 1 - middle image row). Note that the anchor image is the original ROI for the scene to be updated and the overlapping ROIs are only updated where they overlap the anchor ROI. It should also be noted that the Landsat image ROIs all resembled the anchor ROI in shape at the start of the compositing procedure. The resulting cut lines after updating are shown in Fig. 1 (bottom image row). Note that the only cut lines of any of the input images that were updated were those found within the definition domain of the anchor image. Figure 2 shows the resulting image and seam line locations for the given subset of images.



**Fig. 1.** At the top of the figure are found the path/row designations for the subset of Landsat 7 ETM+ scenes shown. The top image row presents the resulting automatically delineated cut lines based on a very large image data set. Scene 196/024 however needed to be updated. Consequently, the middle image row presents the new image to update the data set (196/024) as well as the initial conditions of the neighbouring image ROIs that fall within the definition domain of the input image. Finally, the bottom image row presents the updated seam lines with the new image inserted into the data set. Notice that the cut lines are not the same because the original scene (196/024) was actually wrongly georectified and subsequently a scene acquired on a different date was used to fix the problem.



**Fig. 2.** Eight Landsat 7 ETM+ (band 4 - near infrared) images mosaiced together based on seam lines automatically delineated by the order independent image compositing procedure (shown result after update). The inset presents the puzzle pieces of the mosaic as a reference (path values found at the top and row values to the right). The input images were not enhanced visually in any way and were acquired over a 2 month period during the summer of 2000 over northern Germany.

## 4 Discussion

As can be observed from Fig. 2, it is difficult to see the actual cut lines found in the image mosaic. The reason for this is that the algorithm was devised to cut along salient features found within the imagery. Consequently, the cut lines are mainly found cutting along the boundary between adjacent image objects

that have a large grey level difference. For example, the cut line tends to cut along the shore of a river because the water is dark in the near infrared band used for driving the image composition compared to the river bank, which tends to be brighter. The cut lines are quite indistinguishable even in the absence of any radiometric correction or histogram matching between images. From a human perspective, such an adaptive image cutting is advantageous because it is difficult to tell whether the grey level differences along the cut line are real or not. When the global differences between images are quite marked, i.e., when overlapping images were taken in different seasons or atmospheric conditions, then some image equalisation may be applied.

Another advantage of this image compositing technique is the ability to update certain regions of the image mosaic without the need to re-process the entire data set. As was shown in the results, the new cut lines were delineated based on the new input imagery without affecting those regions that did not require re-processing. This further reinforces the adaptability of the order independent image compositing technique and its ability to focus the delineation of seam lines. Furthermore, the update can include or exclude the original anchor image. The only difference will be the necessity to generate a new overlap matrix that includes the new image to be placed into the composed data set.

Is this the best method for automatically delineating cut lines? Such a question many times has a qualitative answer (i.e., the user makes a judgement) however, to quantify 'best' requires a target to be defined. As Milgram [3] stated, minimising the visual effect of the seam line is the qualitative goal. In the case of the mathematical morphology based image compositing technique, the 'best' criterion could be formulated as the sum of values of the gradient image along the seam line divided by the length of the seam. In other words, the better the seam line, the higher the mean contrast computed along the seam line. Ties can be broken by using the length as a secondary criterion: the shorter the length, the better the seam.

To improve automatic seam line delineation using the above mentioned 'best' criterion, images can first be filtered. To test such an improvement, edge preserving morphological filters with increasing size will be tested and compared to understand their impact on seam line delineation.

One aspect that was not presented in the results was the ability of the order independent image compositing technique to take into account and replace by means of seam lines transient image objects such as clouds. Arbitrarily assigned seam lines cannot take such features into account because they are random. For remotely sensed land applications such objects are not desired, and therefore play an important role in defining seam lines. Taking such an ability into account when delineating seam lines automatically will also influence the criterion for choosing the 'best' possible seam line.

## 5 Conclusions and Future Directions

This paper presented the adaptability of the novel order independent image compositing algorithm based on mathematical morphology to different situations

normally found in remotely sensed applications. Specifically, the ability to automatically delineate a seam line following salient structures which consequently is visually appealing and the ability to easily update seam lines when necessary without reprocessing the entire data set. Updating such information is becoming more and more important as remote sensing image acquisition campaigns continue to provide finer and finer temporal land surface coverage.

This technique recently was also extended to processing Digital Elevation Models (DEM). DEMs generated from ASTER scenes require compositing because they are created from overlapping scenes and contain numerous holes caused by the presence of clouds in the original stereo optical imagery. The final DEM however needs to be continuous. In this case, salient features correspond to ridges.

Processing different regions in parallel in a distributed environment still requires the subsets to be composited. This is possible by adapting the updating procedure to encompass not only a single input scene but a region of overlapping scenes.

## References

1. McNeil, D.: The wet process of laying mosaics. *Photogrammetric Engineering* **15** (1949) 315
2. Wolf, P.: *Elements of Photogrammetry (with Air Photo Interpretation and Remote Sensing)*. McGraw-Hill, New York (1974)
3. Milgram, D.: Adaptive techniques for photomosaicking. *IEEE Transactions on Computers* **26** (1977) 1175–1180
4. Soille, P.: Morphological image compositing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28** (2006) 673–683
5. Price, K.: Computer vision bibliography. <http://iris.usc.edu/Vision-Notes/bibliography/contents.html> (2006)
6. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13** (1991) 583–598
7. Irish, R.: Landsat 7 automatic cloud cover assessment. In Shen, S., Descour, M., eds.: *Proc. of Algorithms for Multispectral, Hyperspectral, and Ultraspectral Imagery VI*. Volume 4049., Society of Photo-Instrumentation Engineers (2000) 348–355
8. Bielski, C., Soille, P.: Order independent image compositing. *Lecture Notes in Computer Science* **3617** (2005) 1076–1083

# Circular Road Signs Recognition with Affine Moment Invariants and the Probabilistic Neural Classifier

Bogusław Cyganek

AGH - University of Science and Technology  
Al. Mickiewicza 30, 30-059 Kraków, Poland  
cyganek@uci.agh.edu.pl

**Abstract.** In this paper the neural classifier for recognition of the circular shaped road signs is presented. This classifier belongs to the road signs recognition module, which in turn is a part of a driver assisting system. The circular shaped prohibition and obligation signs constitute the very important groups within the set of road signs. In this case however, it is not possible for a detector to determine rotation of the shapes that would allow dimension reduction of the search space. Thus the classifier has to be able to properly work with all possible affine deformations. To alleviate this problem we propose to use as features the statistical moments which were shown to be invariant within an affine group of transformations. The classification is performed by the probabilistic neural network which is trained with sign examples extracted from the real traffic scenes. The obtained results show good accuracy of classification and fast operation time.

## 1 Introduction

The idea behind the driving assistance systems (DAS) is to increase security on our roads. Road sign (RS) recognition is a subtask of DAS which purpose is to hint a driver when specific signs are spotted, for instance to limit speed of his or her car [9].

We have designed and built up number of road sign detectors and classification modules, for different groups of shapes [6][7]. In these works, the general idea of the recognition was to register a detected sign to the reference model and then perform recognition step within a group of deformable models [1]. The recognition was done by an ensemble of expert-classifiers (Hamming neural networks - HNN), each operating on a deformable version of the formal road sign data bases (i.e. taken from the law regulations). For a review of the other approaches to the road signs detection and classification please refer to [2][6][7][11][12][20][24].

All road signs are planar shapes that can be subjected to the projective transformation. However, contrary to the triangular or rectangular signs, the problem with circular ones is that their rotation cannot be determined beforehand by the detector. Thus, we cannot limit the search space by one dimension and the classifier has to take into account all possible affine deformations. This results in a prohibitive size of the search space. Therefore some modifications has been proposed in [6] with two systems operating in the spatial and log-polar domains in parallel.

In this paper we take a quite different approach – we collect our road sign data base (RSDB) from real traffic scenes. Then we select circular RSs and compute their affine moment invariants (AMI) [13] which constitute the features which are used for classification. This does not require any registration what speeds up the computation. The recognition is performed by the probabilistic neural net (PNN), previously trained with features collected from the RSDB. The paper presents details of our approach, as well as the experimental results which show good performance of the system.

## 2 System Architecture

Fig. 1. Block diagram of the proposed system for RS recognition depicts basic building blocks of the proposed system with exemplary data at each stage of processing.

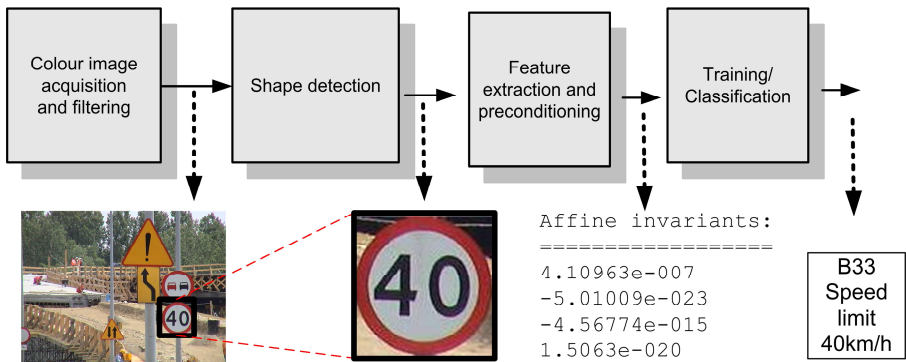


Fig. 1. Block diagram of the proposed system for RS recognition

The first stage consists of the colour image acquisition and filtering. We use the Marlin F-033C camera by Allied Vision Technologies. However, the simpler system with the Logitech internet camera was operating well at good lighting conditions. Then the weighted colour median filter is employed to remove noise and image spikes [8]. The shape detection module operates, first building the colour histogram, then applying the mean-shift algorithm [4]. Next the image is binarized, after which the four AMIs are computed. Finally, these AMIs are normalized to form the feature vectors (see the next section). They are used in the last module – the PNN – during its training and then in the run-time classification.

## 3 Feature Collection – Affine Moment Invariants

The choice of proper features for a classifier has direct influence on the classification results. In our previous approach the circular road signs were binarized. Then a number of deformable models in the spatial and log-polar spaces were created, which constituted prototypes to our classifiers [6]. However, this process involves time consuming image registration procedure. Therefore in the presented approach we

propose to use features which can be directly computed from the detected areas and which are invariant to the affine transformations. This way we circumvent the image registration. Invariance to the affine group is sufficient in our case since the road signs are rigid planar objects.

For a discrete function  $I(x,y)$  defined on the integer grid we can define the moments of order  $(a,b)$  of  $I$ , given as follows:

$$m_{ab} = \sum_{(x,y) \in R} x^a y^b I(x,y). \quad (1)$$

Similarly, the central moments  $c_{ab}$  of order  $(a,b)$  are defined by the following formula:

$$c_{ab} = \sum_{(x,y) \in R} (x - \bar{x})^a (y - \bar{y})^b I(x,y), \quad (2)$$

where the point  $(\bar{x}, \bar{y})$  is called a data centroid. Its coordinates are as follows:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}}, \quad \text{assuming: } m_{00} \neq 0. \quad (3)$$

The AMI were devised from the theory of algebraic moment invariants by Flusser and Suk [13]. They are invariant to the general affine transformation which takes a 2D point  $\mathbf{x}$  into the corresponding  $\hat{\mathbf{x}}$ , in accordance with the following formula:

$$\hat{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}, \quad \text{where } \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}. \quad (4)$$

It was shown [14] that the first four AMIs, given in (5)-(8), are sufficient for reliable pattern recognition.

$$\pi_1 = c_{00}^{-4} (c_{20}c_{02} - c_{11}^2), \quad (5)$$

$$\pi_2 = c_{00}^{-10} (c_{30}^2c_{03}^2 - 6c_{30}c_{21}c_{12}c_{03} + 4c_{30}^3c_{12}^2 + 4c_{03}^3c_{21}^2 - 3c_{21}^2c_{12}^2), \quad (6)$$

$$\pi_3 = c_{00}^{-7} [c_{20} (c_{21}c_{03} - c_{12}^2) - c_{11} (c_{30}c_{03} - c_{21}c_{12}) + c_{02} (c_{30}c_{12} - c_{21}^2)], \quad (7)$$

$$\begin{aligned} \pi_4 = c_{00}^{-11} [ & c_{30}^3c_{03}^2 - 6c_{20}^2c_{11}c_{12}c_{03} - 6c_{20}^2c_{02}c_{21}c_{03} + 9c_{20}^2c_{02}c_{12}^2 + 12c_{20}c_{11}^2c_{21}c_{03} + \\ & + 6c_{20}c_{11}c_{02}c_{30}c_{03} - 18c_{20}c_{11}c_{02}c_{21}c_{12} - 8c_{11}^3c_{30}c_{03} - 6c_{20}c_{02}^2c_{30}c_{12} + \\ & + 9c_{20}c_{02}^2c_{21}^2 + 12c_{11}^2c_{02}c_{30}c_{12} - 6c_{11}c_{02}^2c_{30}c_{21} + c_{02}^3c_{30}^2 ] \end{aligned} \quad (8)$$

In our experiments, prior to computation of AMIs, an image  $I(x,y)$  was binarized, as described in [7]. The crucial part of using the statistical moments for classification is their proper normalization since each moment is characterized by very different exponent (e.g. Fig. 1). In our system, each AMI (5)-(8) was multiplied by a common to its group exponent, so only mantissas were left. Computed this way values are in

the same order of magnitude and if effect have more balanced influence on classification process. After the described normalization,  $\pi_1$ - $\pi_4$  constitute components of the feature vectors to our system.

### 4 Classification with the Probabilistic Neural Network

When the probabilistic density function of data is known then the Bayes classification gives optimal results [10][16]. A set of discriminant functions  $g_i(\mathbf{x})$  is created, where  $i$  is a class label, and, based on a feature vector  $\mathbf{x}$ , an object is classified to a class  $j$  if:

$$\forall_{j \neq i} g_j(\mathbf{x}) > g_i(\mathbf{x}). \tag{9}$$

For the minimum error rate we can take  $g_i(\mathbf{x})=P(\omega_i|\mathbf{x})$ , i.e. the discriminative function is the posterior probability. Further, if we assume that all classes are equally probable, we can use prior probabilities:  $g_i(\mathbf{x})=P(\omega_i)$ . The latter can be estimated from the given populations by nonparametric techniques, e.g. the Parzen windows or its neural realization – the probabilistic neural network (PNN), presented in [21][22][23]. It consists of four layers. The input layer  $\mathbf{X}$  receives input pattern vectors, each of dimension  $n$ . These vectors are normalized. The next layer  $\mathbf{W}$  contains number of weights which store components of the reference patterns. The neurons  $W_{kl}$ , each belonging to only one of the  $p$  classes ( $1 \leq k \leq p$ ), compute a kernel function of its reference pattern and the present input, as follows:

$$W_{kl}(\mathbf{x}) = K\left(\frac{d(\mathbf{x}, \mathbf{x}_{kl})}{h_k}\right), \text{ where } 1 \leq k \leq p, 1 \leq l \leq N_k, \tag{10}$$

$\mathbf{x}_{kl}$  denotes an  $l$ -th pattern from a  $k$ -th class,  $h_k$  is a parameter that controls effective width of the kernel for this class (i.e. its ‘zone of influence’),  $d(.,.)$  denotes a distance between two points in the data space,  $N_k$  denotes number of features in the  $k$ -th class.

Outputs of the neurons from the layer  $\mathbf{W}$  are forwarded into the summation layer, which consists of  $p$  neurons. Output of each is given as follows:

$$g_k(\mathbf{x}) = \alpha_k \sum_{l=1}^{N_k} K\left(\frac{d(\mathbf{x}, \mathbf{x}_{kl})}{h_k}\right), \tag{11}$$

where  $\alpha_k$  is a scaling parameter. The kernel  $K$  plays a role of a weighting function and its choice is independent of the actual, however unknown, pdf of data. Usually it is the Gaussian exponential with the parameter  $\sigma$ . In this case (11) takes on the following form, which was also used in our experiments:

$$g_k(\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{p/2}} \sum_{l=1}^{N_k} e^{-\frac{\|\mathbf{x}-\mathbf{x}_{kl}\|^2}{2\sigma^2}}. \tag{12}$$

The final layer  $\mathbf{M}$  selects one  $g_k(\mathbf{x})$  which gives the maximal response in accordance with (9). Its index  $j$  indicates the class to which the input pattern has been classified by the network:



$$j = \arg \min_{1 \leq k \leq p} \{g_k(\mathbf{x})\}. \tag{13}$$

First training stage of the PNN consists of data normalization which are then used to initialize weights of the  $\mathbf{W}$  layer [10]. However, the second part of the training process concerns determination of the kernel (spread) parameter  $\sigma$  which is not so obvious. Many methods have been proposed for this purpose which, for instance, use of educated guesses, heuristics [18], the Particle Swarm Optimization [15], or the Dynamic Decay Adjustment [3] to name a few. In our experiments we used the two first methods. Specifically, for each population the standard deviation is computed and their mean value is taken as a common  $\sigma$  for the PNN.

Montana proposes a modification to the classic PNN by utilization of the anisotropic Gaussian in (13) and the Atkinson metrics for which the weights are found by the genetic optimization method [19]. This improvement, called the Weigthed PNN, is invariant to affine changes of data in the populations. We plan to explore this direction in future work on this system, since it could allow usage of other features than affine invariants. However, training process can be time demanding in this case.

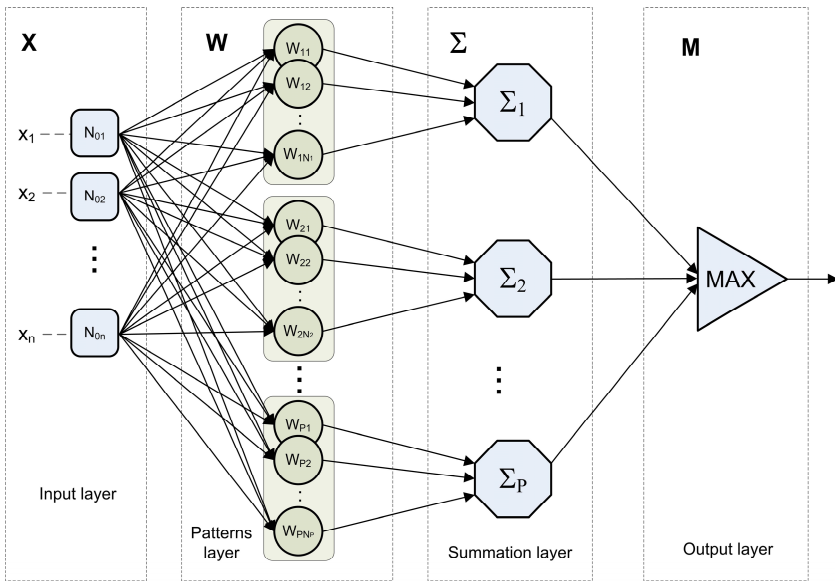


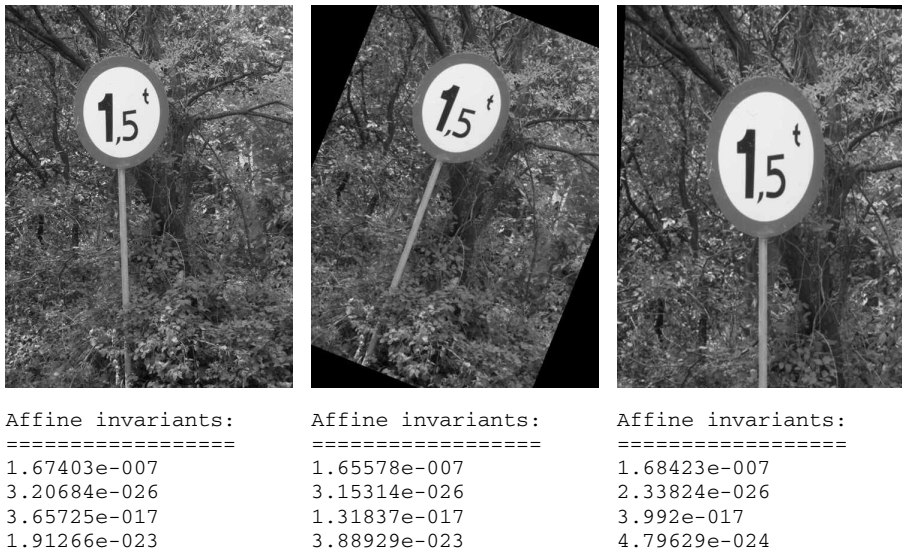
Fig. 2. Structure of the probabilistic neural network used in our system

## 5 Experimental Results

The system was implemented in C++ (Microsoft® Visual C++ 6.0 IDE and Intel C++ compiler 9.0), then tested on the IBM® PC with Pentium 4, 3.4GHz with 2GB RAM.

Fig. 3 presents different affine deformations of a road sign from the “B” group (the prohibitive signs) with the first four AMIs computed for each of the deformations.

It is visible that they are invariant to the rotation, translation and scaling. The small differences are due to non uniform detection of the pictograms. The situation can be improved by the shape registration module [6], however at a cost of additional computations. In practice, however, such big deformations are unusual and for smaller deformation AMIs are sufficient. The second question concerns a choice of the input signal, i.e. whether we should use colour, scalar intensity or binarized versions of a sign. We have not tested the system with colour moments. Instead of using intensity signals, which show high variations over images, we used the binarized version obtained from gray level image. However, for some cases it introduced errors (e.g. the B35-38 signs). In our future work we plan to use the binarization process directly from the colour space to avoid such errors.



**Fig. 3.** Affine moment invariants for different affine transformations of the B18 sign (from left to right): original, 22° rotation, 2° rotation and [0.7,0.8] scaling

Table 1 presents experimental results performed on the selected prohibitive “B” and obligation “C” signs. The signs were collected manually from the traffic scenes encountered on Polish roads. Their data-base (DB) was used to train the PNN classifier. However, the tests in Table 1. Accuracy of recognition for two groups of road signs “B” and “C” under different deformations. The manually extracted data base was used (total of 200 images) were created on DBs which were formed from the original DB by random affine deformations. Lowered precision factor was usually caused by problems with shape selection which influences AMIs. Some false negatives were also detected – this was caused by imperfect detector and the threshold in the two-maximum-separation parameter (described below). Thus, the detector has an influence on the presented parameters.

**Table 1.** Accuracy of recognition for two groups of road signs “B” and “C” under different deformations. The manually extracted data base was used (total of 200 images).

Type of RS Deform. group	“B”		“C”	
	Precision	Recall	Precision	Recall
Rotation up to $\pm 5^\circ$ , scaling $\pm 5\%$	0.81	0.92	0.83	0.94
Rotation up to $\pm 10^\circ$ , scaling $\pm 10\%$	0.70	0.89	0.77	0.93
Rotation up to $\pm 15^\circ$ , scaling $\pm 15\%$	0.69	0.89	0.80	0.90

The classifier always chooses the best class for an input pattern. However, not always it can be a correct answer since the input pattern can belong to false patterns (i.e. not signs). To cope with this situation we have to devise a validation mechanism. However, instead of the fixed threshold value on a classifier response *we compare the two best matches* of a classifier, i.e. the best one, and the second below. If the two are well separated we affirm the answer. We found that if the separation is above 20% then the answer can be correct. The false positives usually have lower separation.

Table 2 contains recognition parameters for the randomly selected real road scenes that contain the signs from the “B” or “C” groups. In this case the classification results are quite promising although some errors in precision and recall are mostly due to noise and occlusions in the input images, which in turn caused some improper sign locations by our detector. Such situation almost always leads to wrong AMIs being computed and, in consequence, worse classification results. However, this does not indicate wrong classification method and can be improved with other detector.

**Table 2.** Accuracy of recognition for two groups “B” and “C” from a hundred of real scenes

“B”		“C”	
Precision	Recall	Precision	Recall
0.71	0.85	0.70	0.91

Some classification errors in the “C” group are also introduced by the signs which differ only by a rotation factor, e.g. the Polish signs C-1, C-3, C-5, C-9, and C-10.

AMIs were also compared with the well known Hu invariants (the first four used) [17]. As expected, the obtained results were worse since the latter cannot deal well with the slanted objects. This observation agrees with the results obtained by Flusser & Suk [14]. Classification times were of order of milliseconds/single image in all tests. However, images with only daily lighting conditions were used.

## 6 Conclusions

The paper presents the system for recognition of the circular road signs. It operates with affine moment invariants, computed from the gray level images. The probabilistic neural network is used for the classification stage. The system complies with the assumptions of the DAS systems, i.e. it classifies different circular RSs encountered

in the real scenes fast and with good quality, it is also simple in implementation and easy for extensions or cooperation with other classifiers.

PNN showed again to be a good choice for pattern classification if we know the classes but don't know their pdf. However, the other classifiers can be used as well, e.g. the self organizing maps, supported vector machines or k-nearest-neighbours. The better results are expected however, when *combining* the presented classifier with the already built 1-NN classifiers with the Hamming NN [6]. Also, when compared with other solutions [6][7], in the case of PNN more problematic is gathering of data.

Special attention has to be devoted to the feature extraction, as well as to the proper choice of the PNN parameter  $\sigma$  in (12) and AMIs preprocessing steps. It turned out that the very important is normalization of the AMIs which tend to show high range of magnitudes.

The experiments show good classification results. Some errors are mostly caused by excessive sign deformations and occlusions which fool our detection system and, in consequence, the classifier. However, the examination of the double response maxima of the classifier allowed greater rejection of false positives. For the future work we plan to improve the detector and built AMIs directly from the colour images.

**Acknowledgement.** This work was supported from the Polish funds for the scientific projects in the year 2007.

## References

1. Amit, Y.: 2D Object Detection and Recognition, MIT Press (2002)
2. Aoyagi, Y., Asakura, T.: A study on traffic sign recognition in scene image using genetic algorithms and neural networks in IEEE Conf. Electronics, Control, (1996) 1838–1843
3. Bertold, M., Diamond, R.: Constructive Training of Probabilistic Neural Networks. *Neurocomputing* 19 (1998) 167-183
4. Bradski, G.: Computer Vision Face Tracking For Use in a Perceptual User Interface. Intel Technical Report (1998)
5. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic Detection and Recognition of Signs From Natural Scenes. *IEEE Trans. on Image Proc.* v.13, no 1 (2004) 87-99
6. Cyganek, B.: Rotation Invariant Recognition of Road Signs with Ensemble of 1-NN Neural Classifiers, LNCS 4132 (2006) 558 – 567
7. Cyganek, B.: Recognition of Road Signs with Mixture of Neural Networks and Arbitration Modules. *Proc. of ISNN, China, LNCS 3973, Springer* (2006) 52 – 57
8. Cyganek, B.: Computational Framework for Family of Order Statistic Filters for Tensor Valued Data, *ICIAR 2006, LNCS 414* (2006) 156 – 162
9. DaimlerChrysler, The Thinking Vehicle, <http://www.daimlerchrysler.com> (2002)
10. Duda, Hart, Stork: *Pattern Classification*. Wiley (2001)
11. Escalera, A., Moreno, L., Salichs, M. A., Armingol, J. M.: Road traffic sign detection and classification, *IEEE Trans. Ind. Electron.*, v. 44, (1997) 848–859
12. Escalera, A., Armingol, J.A.: Visual Sign Information Extraction and Identification by Deformable Models. *IEEE Tr. On Int. Transportation Systems*, v. 5, no 2, (2004) 57-68
13. Flusser, J., Suk, T.: Pattern recognition by affine moments invariants. *Pattern Recognition*, Vol. 13 (1992)

14. Flusser, J., Suk, T.: Affine moments invariants: A new tool for character recognition. *Pattern Recognition Letters*, Vol. 15 (1994) 433-436
15. Georgiou, V.L., Pavlidis, N.G., Parsopoulos, K.E., Alevizos, P.D., Vrahatis, M.N.: Optimizing the Performance of Probabilistic Neural Networks in a Bionformatics Task.
16. Hastie, T., Tibshirani, R., Friedman: *The Elements of Statistical Learning*. Springer (2001)
17. Hu, M.K.: Visual pattern recognition by moment invariants. *IRE Tr. on Inf. Theory* 8 (1962) 179-187
18. Masters, T.: *Practical Neural Network Recipes in C++*, Academic Press (1993)
19. Montana, D.: A Weighted Probabilistic Neural Network. Technical Report (1999)
20. Piccioli, G., Micheli, E.D., Parodi, P., Campani, M.: Robust method for road sign detection and recognition. *Image and Vision Computing*, v.14 (1996) 209-223
21. Rutkowski, L.: *Techniques and Methods of Artificial Intelligence* (in Polish), PWN (2005)
22. Specht, D. F.: Probabilistic neural networks, *Neural Networks*, 1(3) (1990) 109-118
23. Specht, D.F.: Enhancements to Probabilistic Neural Networks, *International Joint Conference on Neural Networks*, Vol. I (1992) 761-768
24. Zheng, Y. J., Ritter, W., Janssen, R.: An adaptive system for traffic sign recognition, in *Proc. IEEE Intelligent Vehicles Symp.* (1994) 165-170

# A Context-Driven Bayesian Classification Method for Eye Location

Eun Jin Koh, Mi Young Nam, and Phill Kyu Rhee

Department of computer science & Engineering Inha University  
Biometric Engineering Research Center Young-Hyun Dong, Incheon, Korea  
supaguri@im.inha.ac.kr, rera@im.inha.ac.kr, phrhee@inha.ac.kr

**Abstract.** In this paper, we present a novel classification method for eye location. It is based on image context analysis. There is general accord that context can be affluent derivation of information about an illumination, character and diversity of object. However, the problem of how to customize contextual influence is not yet solved clearly. Here we describe a naïve probabilistic method for modeling and testing the images of eye patterns. The proposed eye location method employs context-driven adaptive Bayesian framework to relieve the effect due to uneven condition of face image. Based on an easy holistic analysis of face images, the proposed method is able to exactly locate eye position. The experimental results show that the proposed approach can achieve superior performance using various data sets to previously proposed methods.

## 1 Introduction

Face recognition has been a topic of computer vision and one of the most challenging problems for several decades [1]. And face alignment is an important stage in face recognition. Face alignment contain spatial face normalize as scaling and rotating in order to compare with face samples in the database. Usually, face alignment system first operate eye location because attributes of eye is good for locating. Most of the face recognition systems use eye positions which are manually given. There most of real time face recognition system need eye location procedure. In this paper, we propose eye location method using context driven training and testing means. There are three main steps of our eye location system. We will introduce each of these ideas briefly below and then describe them in detail in other sections.

The first step is clustering face images in order to detect eyes in our method. In the real world, there is a strong connection between the objects and environments of the object which is found. It seems that the general visual detection system initially operates context information to arrange characteristics of object. The background can offer the information of object which is attended by human. From environments of object, we can get condition of illumination, noise of image, scale and direction. Because this information affects classification methods or the parameters of system, we employ different training image set, different models and different parameters for each context-cluster. We cluster training face images according to their attributes by

c-means method and extract eye images from each face image. We train the multi-Bayesian classifier with these extracted eye images.

The second step of this paper is a training phase. We train multi-Bayesian classifier using eye images which are extracted by advanced step and non-eye images. The non-eye images are randomly selected by supervised method. The proposed classification system well performs only using eye images, but in order to raise performance of the system, we use also non-eye images.

The third major step of this paper is testing image. This step includes two stages which are clustering face image and classifying search region by multi-Bayesian classifier using model and parameters of specified cluster.

## 2 Image Context-Driven Clustering and Analysis

Image context is any noticeable appropriate attributes that interacts with other images, and image context analysis deals with variation of illumination condition. We use term image context analysis to learn discriminative object patterns which can be measured formally, not intuitively as most previous approaches did. The image context analysis assigns detected face image or manually arranged face image into one of the several image categories. The image contexts are modeled and analyzed by the method of c-means and RBF networks. The RBF networks, just like MLP networks can therefore be used in classification and/or function approximation problems. In the case of a RBF network, we usually prefer the hybrid approach as described below [2]. The RBFs, which have a similar architecture to that of MLPs, however, achieve this goal using a different strategy. One cluster center is updated every time an input vector  $x$  is chosen by c-means from the input data set.

An elementary but popular approximation method shall consider simplifying the computation and accelerating convergence. We tempt to call it the c-means procedure and the goal of this procedure is to find the  $c$  mean vectors  $\mu_1, \mu_2, \dots, \mu_c$ , where  $c$  is the number of cluster centers. This method is popularly known as c-means clustering. We shall follow common equation [6] help to understand c-means manner.

$$\begin{aligned} \hat{P}(\omega_i | x_k, \hat{\theta}) &= \frac{p(x_k | \omega_i, \hat{\theta}) \hat{P}(\omega_i)}{\sum_{j=1}^c p(x_k | \omega_j, \hat{\theta}_j) \hat{P}(\omega_j)} \\ &= \frac{|\hat{\Sigma}_i|^{-1/2} \exp\left[-\frac{1}{2}(x_k - \hat{\mu}_i)' \hat{\Sigma}_i^{-1} (x_k - \hat{\mu}_i)\right] \hat{P}(\omega_i)}{\sum_{j=1}^c |\hat{\Sigma}_j|^{-1/2} \exp\left[-\frac{1}{2}(x_k - \hat{\mu}_j)' \hat{\Sigma}_j^{-1} (x_k - \hat{\mu}_j)\right] \hat{P}(\omega_j)}. \end{aligned} \tag{1}$$

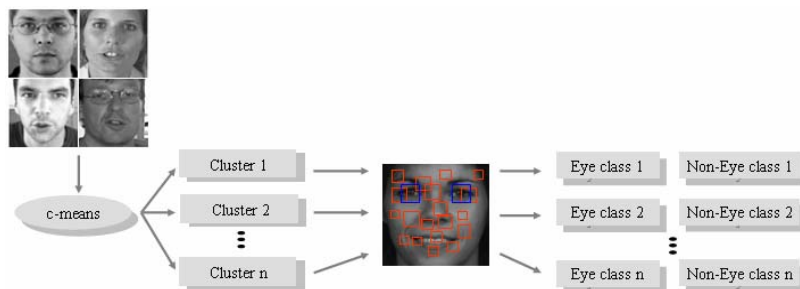
From (1), it is clear that the probability  $\hat{P}(\omega_i | x_k, \hat{\theta})$  is large when the squared Mahalanobis distance  $(x_k - \hat{\mu}_i)' \hat{\Sigma}_i^{-1} (x_k - \hat{\mu}_i)$  is small. Suppose that we merely compute the squared Euclidean distance  $\|x_k - \hat{\mu}_i\|^2$ , find the mean  $\hat{\mu}_m$  nearest to and approximate  $\hat{P}(\omega_i | x_k, \hat{\theta})$  as

$$\hat{P}(\omega_i | x_k, \hat{\theta}) \equiv \begin{cases} 1 & \text{if } i = m \\ 0 & \text{otherwise.} \end{cases}$$

Then the iterative application of  $\hat{\mu}_i$  leads to the following procedure for finding  $\mu_1, \mu_2, \dots, \mu_c$ . In the absence of other information, we may need to guess the “proper” number of clusters,  $n$ . Likewise; we may assign  $n$  based on the final application. Here and throughout, we denote the known number of patterns as  $m$ , and the desired number of cluster  $n$ . It is tradition to let  $n$  samples chosen randomly from the data set serve as initial cluster centers. Then the algorithm is:

- Step 1** *begin initialize*  $n, m, \mu_1, \mu_2, \dots, \mu_c$   
**Step 2** *do* classify  $n$  samples according to nearest  $\mu_i$   
**Step 3** *recompute*  $\mu_i$   
**Step 4** *until* no change in  $\mu_i$   
**Step 5** *return*  $\mu_1, \mu_2, \dots, \mu_c$   
**Step 6** *end*

In training phase, we classify face images along illumination peculiarity whereas in experience phase, we apply Bayesian theory with optimized different thresholds at each cluster. We call this method a context-driven. The general Bayesian method [3] applies same threshold to whole test images, but the context-driven method applies several thresholds according to the number of clusters.



**Fig. 1.** The basic training algorithm

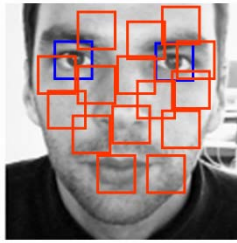
As shown in Fig. 1, the idea is to train the network in two separate stages. At first stage, we perform an unsupervised training using c-means. And at the second stage, the diverse thresholds are trained using the regular supervised approach. Input pattern is vectorized for grayscale image size of  $16 \times 16$  pixels. RBF network has an architecture that is very similar to the traditional three-layer back-propagation network. But, in RBF, the transformation from the input space to the hidden unit space is non-linear, whereas the transformation from the hidden-unit space to the output-space is linear. And RBF classifier expands input vectors into a high dimensional space. In this paper, hidden units are trained using k-means. The network input consists of  $n$  normalized and rescaled size of  $1/2$  face k-images fed to the network as 1 dimension vector. And input unit has floating value  $[0, 1]$ . The vector value is normalized.



### 3 Training the Multi-Bayesian Classifier

After clustering sample images according to their attributes, we can get  $n$  face image clusters. In order to train the multi-Bayesian classifier, we need extract eye images and non-eye images for each cluster. The extracted eye images from cluster  $i$  are normalized by 16x16 pixels, and non-eye images are randomly extracted from cluster. We collect 1024 faces (contain 2048 eyes) from FERET gallery dataset for modeling eye class and extract eye regions manually. The eye regions are resized as 16x16 pixels and converted to vector by 2D Haar transform.

An example of extraction is shown in Fig. 2. Blue regions indicate areas which are extracted as eye images and red regions are extracted as non-eye images.



**Fig. 2.** An example of extraction. Blue regions are extracted as eye images and red regions are randomly extracted from face image as non-eyes.

The multi-Bayesian classifier consists of sets of clusters, eye models and non-eye models. In order to enhance the robustness of the representation to noise and diversity of eye around, the system employs context-driven classification strategy. The multi-Bayesian can be described by an ordered triple data model defined as multi Bayesian classifier :  $B = (C, E, R)$ , where  $C = \{c_1, \dots, c_n\}$  is the set of clusters of the face images,  $E = \{e_1, \dots, e_n\}$  is the Bayesian model of eye class, and  $R = \{r_1, \dots, r_n\}$  is the Bayesian model of non-eye class. The eye class model,  $e_i$  represents model which is made up of eye images extracted from face image of cluster  $i$ , the non-eye class model,  $r_i$  describes model which consists of non-eye images extracted from face image of cluster  $i$ .

#### 3.1 Class Modeling for Multi-Bayesian Classifier

The posterior probability density of the eye class of cluster  $i$ ,  $e_i$ , is modeled as a normal distribution [3]:

$$P(x | e_i) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - m_i)^t \sum_i^{-1} (x - m_i) \right\}, \quad (2)$$

where  $m_i$  and  $\sum_i$  are the mean and the covariance matrix of eye images. The covariance matrix,  $\sum_i$ , can be factorized into the following form by principal component analysis (PCA).

$$\Sigma_i = E_i R_i E_i^t \quad (3)$$

with

$$E_i E_i^t = \text{diag} \{ \lambda_1, \lambda_2, \dots, \lambda_n \}, \quad (4)$$

where is an orthogonal eigenvector matrix,  $R_i$  is a diagonal eigenvalue matrix with diagonal elements i.e., eigenvalues, in dwindling order ( $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ ). A significant attribute of PCA is its optimal signal recomposition in the sense of lowest mean-square error when only a subset of principal components is used to depict the earlier signal [2]. The principal components are specified by the following vector,  $X$ ,

$$X = E_e^t (x - m_e). \quad (5)$$

Note that the components of  $X$  are the principal components. Applying the optimal signal recomposition attribute of PCA, we use only the first  $m$  ( $m \ll n$ ) principal components to estimate the posterior density function. Some of training images which are used to construct eye class model can be seen in Fig. 2.

Practically any image can be offered as a non-eye sample because the domain of non-eye samples is much wider than the domain of eye examples. However, selecting a "representative" set of non-eye is troublesome task. An ideal situation is that the non-eye samples are similar to eyes but not. The non-eye class modeling starts with extracting non-eye samples that do not contain any whole eye region at all as red region of Fig. 2. Then we generate "representative" non-eye samples by applying (6) to extracted images. Those representative samples that lie closest to the eye class are chosen as modeling samples for the estimation of the posterior density function of the non-eye class which is also modeled similar to manner of eye class.

Notion of distance is considered necessary in order to design righteous model. In this paper, we are with Mahalanobis distance instead of Euclidean distance. A common usage of Mahalanobis distance is for comparing feature vectors whose elements are quantities having different ranges and amounts of variation. It is based on correlations between variables by which different patterns and feature can be identified, analyzed or tested. This strategy is very useful tools for determining similarity of unknown test sample with a known set. Mahalanobis distance differs from Euclidean distance in that it considers the correlations of the data set. Mathematically, the Mahalanobis distance from a group of values with mean  $m = (m_1, m_2, m_3, \dots, m_n)$  and covariance matrix  $\Sigma$  for a multivariate vector  $I = (I_1, I_2, I_3, \dots, I_n)$  is defined as:

$$d(x) = \sqrt{(x - m)^t \Sigma^{-1} (x - m)}. \quad (6)$$

For the multi-Bayesian classification, let  $d_c(x)_e$  be the Mahalanobis distance between the pattern of the region of interest and eye class of cluster  $c$ , and  $d_c(x)_n$  be that of non-eye class of cluster  $c$ . The distances,  $d_c(x)_e$  and  $d_c(x)_n$  can be computed from the input pattern  $x$ . To classify classes, we use two thresholds,  $\theta$  and  $\tau$  as follows:

$$\begin{aligned} \theta &= \max(d_c(\alpha)_e) \\ \tau &= \max(d_c(\beta)_e - (d_c(\beta)_n)), \end{aligned} \tag{7}$$

where  $\alpha$  is the pattern of training sample image of eye class and  $\beta$  is the pattern of training sample image of non-eye class. Because the two thresholds are constant values, they are computed in the training time.

### 4 Multi-Bayesian Discriminate Method

The first part of our multi-Bayesian discriminate method is an image context-driven clustering method that clusters an input face image as a specified cluster  $c$  and in order to signify the presence or absence of an eye, we use Bayesian classifier  $c$  which has two models made up of eye model and non-eye model. The second part of our method is a multi-resolution Bayesian classifier that locates eye in the face image. We use the window that acquires as input a 16x16 pixel area of the image and is classified as eye or non-eye. For robust precise location, the window is applied at every pixel in the face image. In order to locate eyes larger than the window size, the input face image is decreased in size, and the window is applied at same size. For the system which is introduced here, we scale repeatedly the image down with a factor of 1.3 until its size becomes smaller than window size because we don't know how large are eye regions before detect eyes. The basic concept of proposed method is shown in Fig. 3. After clustering many face images, we make eye and non-eye statistical models using each cluster's face images. With these models, we can decide which image region contains eye.

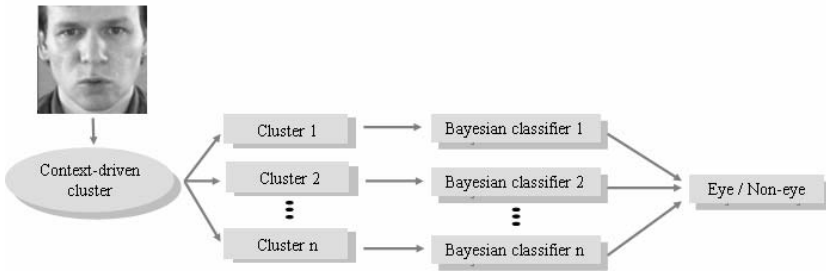


Fig. 3. The basic classification algorithm

The multi-Bayesian classifier offers the classifying rule to the eye location system as follows:

$$x \in \begin{cases} w_e & \text{if } (d(x)_e < \theta) \text{ and } (d(x)_e - d(x)_n < \tau) \\ w_n & \text{otherwise,} \end{cases} \tag{8}$$

where  $w_e$  is eye class and  $w_n$  is non-eye class.

## 5 Experimentation

There are several manners to measure the accuracy of eye location used in previous researches [4]. We adopt a measurement which is independent of scale, called relative accuracy, to estimate the accuracy of eye location [4]. We define an ideal eye position as a center of the pupil of the located eye. The relative accuracy of eye location is defined as follows [4]:

$$error = \frac{\max(d_l, d_r)}{d_{lr}}, \quad (9)$$

where  $d_l$  and  $d_r$  are error distance of right eye and left eye, and  $d_{lr}$  is distance between left eye and right eye.

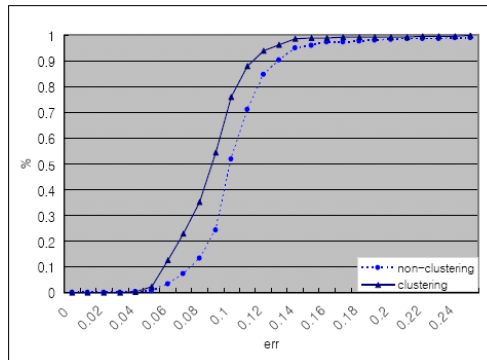
**Table 1.** Comparative performance of multi-Bayesian method and general Bayesian method ( $n=3$ )

classes	images	general Bayesian method			context-driven method		
		detect	false	rate	detect	False	rate
class 0	293	265	28	90.44	238	9	96.93
class 1	672	657	15	97.77	667	5	99.26
class 2	128	116	12	90.63	127	1	99.22
total	1093	1038	55	94.97	1078	15	98.63

$err < 0.14$

To make comparison with multi-Bayesian method with general Bayesian method, we cluster 1093 images from FERET *fa* set for three. Performances of the methods are demonstrated in Table 1 and Fig. 4. They show that multi-Bayesian method is superior to general Bayesian method. From the table, we can conclude that c-means based multi-Bayesian method have an effect on eye location.

The proposed system is compared with other systems. We select paper [4] and [5] as targets of comparison because they use same evaluation protocol (*err*) and similar



**Fig. 4.** Comparative curves between clustering and non-clustering method

data sets with us. In paper [5], the detection rate is 94.81% at BioID under  $\text{err} < 0.25$ . On the other hand, our detection rate is 99.91% if  $\text{err} < 0.25$ . And their system achieve on 97.18% of JAFFE data set. But backgrounds and illumination conditions of JAFFE are not as complex and diverse as these of FERET. In paper [4], the detection is considered to be correct if  $\text{err} < 0.20$ . Their detection rate is 99.1% but they did not indicate its data set. We evaluate proposed method under the same evaluation protocol. The detection rate of our system is 99.27% if  $\text{err} < 0.17$ , and the detection rate is 99.91 % if  $\text{err} < 0.25$ .

We offer some examples out of the test sets for visual examination at Fig. 5. The system appears to be robust to the presence of glasses, closed eyes, slightly rotated faces and even significant pose changes.



Fig. 5. Some eye location examples

## 6 Conclusion

This paper describes a novel multi-Bayesian discriminate method for robust eye location. The novelty of this paper comes from the clustering faces, the statistical modeling of eye and non-eye classes, and the method of determining thresholds. First, the system use clustering method. Second, statistical modeling evaluates the posterior probability density functions of the eye and non-eye classes. Finally, we adopt peculiar method of determining which can classify eyes and non-eyes. Experimental results show multi-Bayesian method is effect on pattern classification. The multi-Bayesian classifier achieves 98.63 percent eye location accuracy under  $\text{error} < 0.14$ .

In addition, because the clustering and multi-Bayesian discriminate method don't localized for case of eye, the proposed method in this paper totally can be applied for location of other face organs such as nose or mouth.

## References

1. W. Zhao, R. Chellappa, J. Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey," *ACM Computing Surveys*, vol. 35, no. 4, (2003).
2. M. H. Hassoun. "Fundamentals of Artificial Neural Networks," MIT Press, (1995).
3. C. Liu, "A Bayesian Discriminating Features Method for Face Detection" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, (2003) pp. 725-740.
4. Y. Ma, X. Ding, Z. Wang, N. Wang, "Robust precise eye location under probabilistic framework", *IEEE International Conference on Automatic Face and Gesture Recognition*, (2004)
5. H. Zhou, X. Geng, "Projection functions for eye detection," *Pattern Recognition*, (2004), in press.
6. Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification 2/e", Wiley-interscience press, (2000).

# Computer-Aided Vision System for Surface Blemish Detection of LED Chips

Hong-Dar Lin<sup>1</sup>, Chung-Yu Chung<sup>1</sup>, and Singa Wang Chiu<sup>2</sup>

<sup>1</sup> Department of Industrial Engineering and Management,

<sup>2</sup> Department of Business Administration,

Chaoyang University of Technology,

168 Jifong E. Rd., Wufong Township, Taichung County 41349, Taiwan (R.O.C.)

hdlin@cyut.edu.tw

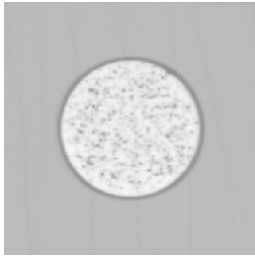
**Abstract.** This research explores the automated detection of surface blemishes in light-emitting diode (LED) chips. An LED is a semiconductor device that emits visible light when an electric current passes through the semiconductor chip. Water-drop blemishes, commonly appearing on the surfaces of chips, impair the appearance of LEDs as well as their functionality and security. Consequently, detecting water-drop blemishes becomes crucial for the quality control of LED products. We first use the one-level Haar wavelet transform to decompose a chip image and extract four wavelet characteristics. Then, the  $T^2$  statistic of multivariate statistical analysis is applied to integrate the multiple wavelet characteristics. Finally, the wavelet based multivariate statistical approach judges the existence of water-drop blemishes. Experimental results show that the proposed method achieves an above 95% detection rate and a below 1.5% false alarm rate in inspecting water-drop blemishes of LED chips.

## 1 Introduction

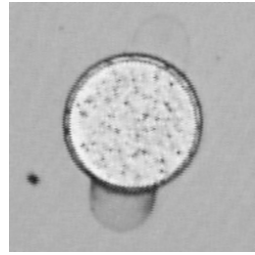
A light-emitting diode (LED) is a semiconductor device that emits visible light when an electric current passes through the semiconductor chip. Compared with incandescent and fluorescent illuminating devices, LEDs have lower power requirement, higher efficiency, and longer lifetime. Typical applications of LED components include indicator lights, LCD panel backlighting, fiber optic data transmission, etc. To meet consumer and industry needs, LED products are being made in smaller sizes, which increase difficulties of product inspection.

Surface defects impair the appearance of LEDs as well as their functionality and security. As inspecting surface defects by human eyes is ineffective and inefficient, this research aims to develop an automated vision system for detecting one common type of LED surface defects, water-drop blemishes formed by the steam generated during the production process. Difficulties exist in automatically inspecting water-drop blemishes because of their semi-opaque appearance and the low intensity contrast between their surface and the rough exterior of a LED chip. Figure 1 displays the LED chip images with and without water-drop blemishes.

Defect detection techniques, generally classified into the spatial domain and the frequency domain, compute a set of textural features in a sliding window and search



(a) LED chip without defect



(b) LED chip with water-drop defects

**Fig. 1.** LED chip images

for significant local deviations among the feature values. Siew et al. [1] apply the co-occurrence matrix method, a traditional spatial domain technique, to assess carpet wear by using two-order gray level statistics to build up probability density functions of intensity changes. For another spatial domain example, Latif-Amet et al. [2] present wavelet theory and co-occurrence matrices for detection of defects encountered in textile images and classify each sub-window as defective or non-defective with a Mahalanobis distance.

As to techniques in the frequency domain, Tsai and Hsiao [3] propose a multiresolution approach for inspecting local defects embedded in homogeneous textured surfaces. By properly selecting the smooth subimage or the combination of detail subimages in different decomposition levels for backward wavelet transform, regular, repetitive texture patterns can be removed and only local anomalies are enhanced in the reconstructed image. Also, Tsai and Wu [4] adopt Gabor transform to determine three texture features (scale, frequency and orientation) and use Gabor energy differences to discriminate defect locations.

Regarding defect detection applications in the electronic industry, Jiang et al. [5] use luminance measurement equipment to collect data of MURA-type defects of Liquid Crystal Display (LCD) surfaces, and apply analysis of variance and exponentially weighted moving average techniques to develop an automatic inspection procedure. For the same LCD surface defect detection, Lin and Chiu [6] use multivariate Hotelling  $T^2$  statistic to integrate different coordinates of color models and constructs a  $T^2$  energy diagram to represent the degree of color variations for selecting suspected defect regions. Then, an Ant Colony based approach that integrates computer vision techniques precisely identifies the flaw point defects in the  $T^2$  energy diagram. The Back Propagation Neural Network model determines the regions of small color variation defects based on the  $T^2$  energy values. Furthermore, in the recent decade, many vision systems have been developed for the inspection of surface defects on semiconductor wafers [7]-[8]. For instance, Fadzil and Weng [9] implement a vision inspection system that achieves a 90% probability of accurately classifying defects, scratches, contamination, blemishes, off center defects, etc. in the encapsulations of diffused LED products.

The aforementioned techniques perform well in anomaly detection, but most of them do not detect defects with the properties of water-drop blemishes [2]-[9]. This research has been motivated by the need for an efficient and effective technique that

detects and locates semi-opaque and low-contrast water-drop blemishes on random texture surfaces.

## 2 Proposed Method

To detect water-drop blemishes of LED chips, this research adopts the one-level Haar wavelet transform to conduct image pre-processing and extract wavelet characteristics. We apply the Hotelling  $T^2$  statistic of multivariate statistical analysis to integrate multiple wavelet characteristics and then develop the wavelet based multivariate statistical approach to judge the existence of water-drop blemishes in an image.

### 2.1 Wavelet Decomposition and Characteristics

Wavelet transform provides a convenient way to obtain a multiresolution representation, from which texture features can be easily extracted. The merits of using wavelet transform include local image processing, simple calculations, high speed processing and multiple image information [10]-[12]. The Haar wavelet transform is one of the simplest and basic wavelet transformations [12]. A standard decomposition of a two-dimensional image can be done by first applying the 1-D Haar wavelet transform to each row of pixel values, treating these transformed rows as if they were themselves an image, and then performing another 1-D wavelet transform to each column. The Haar transform can be computed stepwise by the mean value and half of the differences of the tristimulus values of two contiguous pixels. Based on the transfer concept of the 1-D space, the Haar wavelet transform can process a 2-D image of ( $M \times N$ ) pixels in the following way:

$$\begin{array}{l}
 \text{Row transfer:} \\
 \left\{ \begin{array}{l} f_R(i, j) = \left[ \frac{f(i, 2j) + f(i, 2j+1)}{2} \right], \\ f_R(i, j + \left[ \frac{N}{2} \right]) = \left[ \frac{f(i, 2j) - f(i, 2j+1)}{2} \right], \\ \text{where } 0 \leq i \leq (M-1), 0 \leq j \leq \left[ \frac{N}{2} \right] - 1, [ \ ] \text{ is Gauss symbol.} \end{array} \right. \\
 \text{Column transfer:} \\
 \left\{ \begin{array}{l} f_C(i, j) = \left[ \frac{f_R(2i, j) + f_R(2i+1, j)}{2} \right], \\ f_C(i + \left[ \frac{M}{2} \right], j) = \left[ \frac{f_R(2i, j) - f_R(2i+1, j)}{2} \right], \\ \text{where } 0 \leq i \leq \left[ \frac{M}{2} \right] - 1, 0 \leq j \leq (N-1). \end{array} \right. \quad (1)
 \end{array}$$

In the above expressions of (1),  $f(i, j)$  represents an original image,  $f_R(i, j)$  the row transfer function of  $f(i, j)$ , and  $f_C(i, j)$  the column transfer function of  $f_R(i, j)$ . As  $f_C(i, j)$  is also the outcome of the wavelet decomposition of  $f(i, j)$ , the outcomes of a wavelet transform can be defined as:

$$\left\{ \begin{array}{l} A(i, j) = f_C(i, j), \quad D_1(i, j) = f_C(i, j + \left[ \frac{N}{2} \right]), \\ D_2(i, j) = f_C(i + \left[ \frac{M}{2} \right], j), \quad D_3(i, j) = f_C(i + \left[ \frac{M}{2} \right], j + \left[ \frac{N}{2} \right]), \\ \text{where } 0 \leq i \leq \left[ \frac{M}{2} \right] - 1, 0 \leq j \leq \left[ \frac{N}{2} \right] - 1. \end{array} \right. \quad (2)$$



One level of wavelet decomposition generates one smooth subimage and three detail subimages that contain fine structures with horizontal, vertical, and diagonal orientations. An image is decomposed by wavelet transform into one approximation subimage ( $A$ ) and three detail subimages ( $D_1, D_2$  and  $D_3$ ). These four subimages, each of which has a size of  $(M/2 \times N/2)$  pixels, form the wavelet characteristics.

### 2.2 Wavelet Based Multivariate Statistical Approach

The wavelet based multivariate approach decomposes an image of  $(M \times N)$  pixels into a set of subimages, each of which has a size of  $(m \times n)$  pixels and is a multivariate processing unit. The original image has  $g \times h$  (i.e.  $M/m \times N/n$ ) multivariate processing units, each of which can be further decomposed into  $a \times b$  wavelet processing units. For each wavelet processing unit, the wavelet transform can be applied to the region of  $(m/a \times n/b)$  pixels to obtain four wavelet characteristics  $A, D_1, D_2$  and  $D_3$  through calculations. The multivariate statistic  $T^2$  integrates the multiple wavelet characteristics into a  $T^2$  value for each multivariate processing unit. This  $T^2$  value can be regarded as a distance value of a multivariate processing unit. The larger the  $T^2$  distance value, the more distinctive the region is from the normal area. Thus, the more easily the region can be judged as defective.

The proposed wavelet based approach assumes that the size of a multivariate processing unit is  $4 \times 4$  (i.e.  $m \times n$ ) pixels and the size of a wavelet processing unit is  $2 \times 2$  (i.e.  $a \times b$ ) pixels. One multivariate processing unit will have  $2 \times 2$  (i.e.  $m/a \times n/b$ ) wavelet processing units. That is, four wavelet processing units  $C(x_a, y_b)$  can be defined as one multivariate processing unit  $M(x, y)$ , where  $a$  and  $b$  are integers and  $(1 \leq a, b \leq 2)$ . The corresponding spatial coordinates of  $C(x_a, y_b)$  are a square with size  $2 \times 2$  pixels from  $f(4 * x + a, 4 * y + b)$  to  $f(4 * x + a + 1, 4 * y + b + 1)$ . Thus, one  $M(x, y)$  includes four  $C(x_a, y_b)$ , which are  $C(x_1, y_1), C(x_1, y_2), C(x_2, y_1)$  and  $C(x_2, y_2)$ . One  $C(x_a, y_b)$  can be decomposed by wavelet transform to obtain one approximated characteristic  $A(x_a, y_b)$  and three detail characteristics  $D_1(x_a, y_b), D_2(x_a, y_b)$  and  $D_3(x_a, y_b)$ .

The calculation formulas of a multivariate control procedure [13]-[15] can be rewritten as the following equations of (3) to (8) to represent a multivariate process of images:

$$\bar{X}_{M(x,y)} = \left[ \frac{1}{a \times b} \sum_{i=1}^a \sum_{j=1}^b X_{C(x_i, y_j), p} \right]_{p \times 1}, \tag{3}$$

$$\bar{\bar{X}} = \left[ \frac{1}{g \times h} \sum_{i=0}^{g-1} \sum_{j=0}^{h-1} \bar{X}_{M(i, j), p} \right]_{p \times 1}, \tag{4}$$

$$S_{M(x,y), p}^2 = \frac{1}{a \times b - 1} \sum_{i=1}^a \sum_{j=1}^b \left( X_{C(x_i, y_j), p} - \bar{X}_{M(x,y), p} \right)^2, \tag{5}$$

$$S_{M(x,y), p,q} = \frac{1}{a \times b - 1} \sum_{i=1}^a \sum_{j=1}^b \left( X_{C(x_i, y_j), p} - \bar{X}_{M(x,y), p} \right) \left( X_{C(x_i, y_j), q} - \bar{X}_{M(x,y), q} \right), \tag{6}$$

$$S_p^2 = \frac{1}{g \times h} \sum_{i=0}^{g-1} \sum_{j=0}^{h-1} S_{M(i,j),p}^2 \tag{7}$$

$$S_{p,q} = \frac{1}{g \times h} \sum_{i=0}^{g-1} \sum_{j=0}^{h-1} S_{M(i,j),p,q} \tag{8}$$

where  $X_{C(x_a, y_b), p}$  is the  $p$ -th image characteristic of a wavelet processing unit  $C(x_a, y_b)$ ;  $\bar{X}_{M(x, y)}$  is the mean matrix of image characteristics in a multivariate processing unit  $M(x, y)$ ;  $\bar{X}_{M(i, j), p}$  is the mean value of the  $p$ -th image characteristic of  $M(i, j)$ ;  $S_{M(x, y), p}^2$  is the variance of the  $p$ -th image characteristic of  $M(x, y)$ ;  $S_{M(x, y), p, q}$  is the covariance of the  $p$ -th and the  $q$ -th image characteristics of  $M(x, y)$ . The multivariate matrices used in this research can be expressed as follows:

$$X_{C(x_a, y_b)} = \begin{bmatrix} A(x_a, y_b) \\ D_1(x_a, y_b) \\ D_2(x_a, y_b) \\ D_3(x_a, y_b) \end{bmatrix}, \quad \bar{X}_{M(x, y)} = \begin{bmatrix} \bar{A}(x, y) \\ \bar{D}_1(x, y) \\ \bar{D}_2(x, y) \\ \bar{D}_3(x, y) \end{bmatrix}_{4 \times 1} = \begin{bmatrix} \frac{1}{a \times b} \sum_{i=1}^a \sum_{j=1}^b A(x_i, y_j) \\ \frac{1}{a \times b} \sum_{i=1}^a \sum_{j=1}^b D_1(x_i, y_j) \\ \frac{1}{a \times b} \sum_{i=1}^a \sum_{j=1}^b D_2(x_i, y_j) \\ \frac{1}{a \times b} \sum_{i=1}^a \sum_{j=1}^b D_3(x_i, y_j) \end{bmatrix}_{4 \times 1} \tag{9}$$

Normal texture images are used to estimate the parameters of standard texture characteristics. The sample mean matrix ( $a \times b$ ) and the sample covariance matrix ( $S$ ) describe the properties and relations between normal and defect images. The  $S_{M(x, y), p, q}$  and  $S$  are defined as:

$$\bar{X} = \begin{bmatrix} \bar{A} \\ \bar{D}_1 \\ \bar{D}_2 \\ \bar{D}_3 \end{bmatrix}_{4 \times 1}, \quad S = \begin{bmatrix} S_A^2 & S_{A, D_1} & S_{A, D_2} & S_{A, D_3} \\ S_{D_1, A} & S_{D_1}^2 & S_{D_1, D_2} & S_{D_1, D_3} \\ S_{D_2, A} & S_{D_2, D_1} & S_{D_2}^2 & S_{D_2, D_3} \\ S_{D_3, A} & S_{D_3, D_1} & S_{D_3, D_2} & S_{D_3}^2 \end{bmatrix}_{4 \times 4} \tag{10}$$

where  $S_p^2$  is the sample variance of the  $p$ -th wavelet characteristic of an image;  $S_{p, q}$  is the sample covariance of the  $p$ -th and the  $q$ -th wavelet characteristics of an image.

The  $T^2$  distance value of the multivariate processing unit  $M(x, y)$  of a testing image can be defined as:

$$T_{M(x, y)}^2 = a \times b \left[ \bar{X}_{M(x, y)} - \bar{\bar{X}} \right]' S^{-1} \left[ \bar{X}_{M(x, y)} - \bar{\bar{X}} \right] \tag{11}$$

where  $a \times b$  is the number of wavelet process units in a multivariate processing unit.  $\bar{X}_{M(x, y)}$  is the mean matrix of image characteristics in the multivariate processing unit of a testing image.  $\bar{\bar{X}}$  and  $S$  are respectively the mean matrix and the covariance matrix of image characteristics of a normal image. The control limit is as follows:

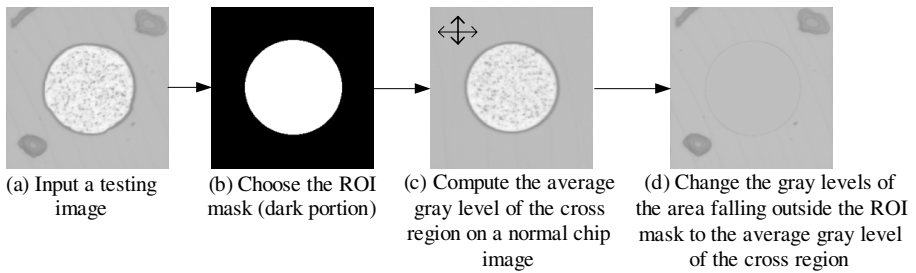
$$\frac{p(m-1)(n-1)}{mn-m-p+1} F_{\phi, p, (mn-m-p+1)}, \tag{12}$$

where  $F$  is a tabulated value of the  $F$  distribution at the significance level of  $\phi$ .

### 3 Experimental Results

Experiments are conducted on real LED chips to evaluate the performance of the proposed approach. We test 120 LED images, of which 25 have no defects and 95 have various water-drop defects. After conducting different experiments, we found the most appropriate size of a multivariate processing unit to be 4 x 4 pixels. At this size, the proposed approach achieves the best performance considering the sample training time, the recognition time of the testing period, the size of the defect area and other factors in the multivariate processing.

To maximize the number of chips on a wafer, every chip is located very close to its neighboring chips. As the carrier plate moves to have the image of the next chip captured, the movement might cause the CCD to deviate from its original position and the image capturing device to vibrate. Thus, the images of all the chips might be captured with slight differences. That is, not all the chips are located in the exactly same positions in their individual images. As a result, one ROI (region of interest) mask is needed for each image to specify the locations in which water-drop blemishes may possibly exist. Figure 2 presents the procedure of removing the central portion of a defective LED chip.



**Fig. 2.** The procedure of removing the central portion of a defective LED chip

Figure 3 shows partial results of detecting water-drop defects by the Otsu method [16], the Iterative method [17], and the proposed wavelet based multivariate statistical approach, individually. The wavelet based  $T^2$  method detects most of the water-drop blemishes while the Otsu and the Iterative methods miss some defect regions. The performance evaluation indices,  $(1-\alpha)$  and  $(1-\beta)$ , are used to represent correct detection judgments; the higher the two indices, the more accurate the detection results. The type I error  $\alpha$  is the probability of incorrectly judging the normal regions as defects. The type II error  $\beta$  represents the probability of failing to alarm real defects. Figure 4 depicts the detection results of the wavelet based  $T^2$  method at different

significant levels. The wavelet-based  $T^2$  method performs well in detecting water-drop blemishes ( $(1-\beta) > 95\%$ ) at a significance level of above 0.000001. The average detection rates of all testing samples by the three methods are, respectively, 98.5% (the wavelet based  $T^2$  method), 83.2% (the Iterative method), 82.4% (the Otsu method). The proposed wavelet based multivariate approach has higher detection rates than do either of the two traditional methods applied to LED chip images. The wavelet based  $T^2$  method excels in its ability of correctly discriminating water-drop blemishes from normal regions.

As the decision threshold value changes, so do its false alarm rate ( $\alpha$ ) and detection rate ( $1-\beta$ ), both of which are used to describe the performance of a test according to

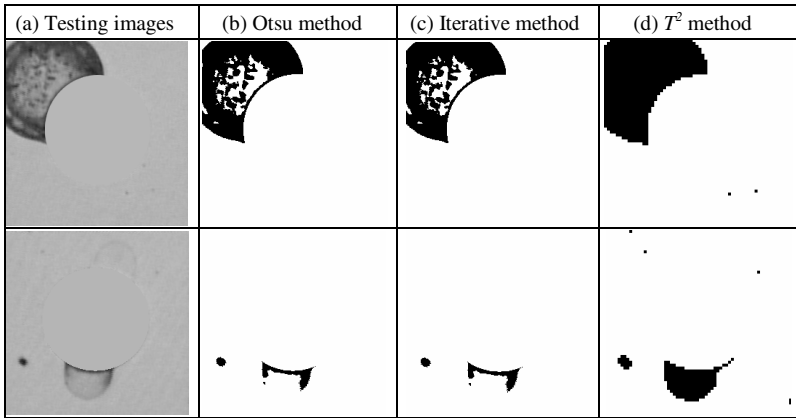


Fig. 3. Partial detection results of Otsu, Iterative and wavelet based  $T^2$  methods

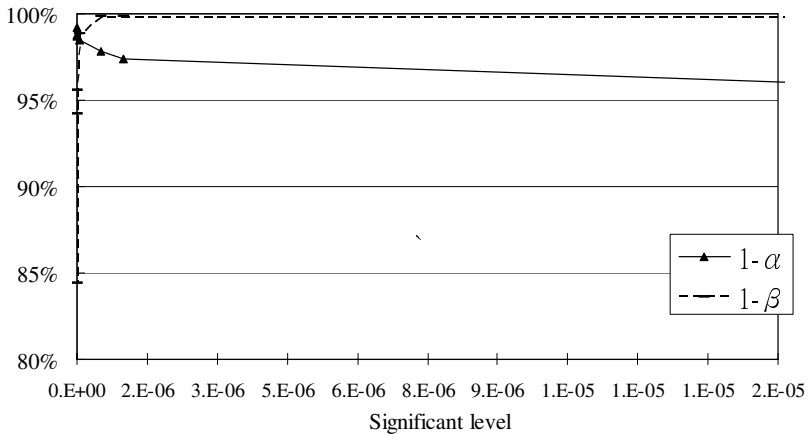


Fig. 4. Detection results of the proposed T2 method at different significant levels

hypothesis testing theory [18]. When various decision thresholds are used, their pairs of false alarm rates and detection rates are plotted as points on a Receiver Operating Characteristic (ROC) curve. The ROC curve of the wavelet based  $T^2$  method and the two points of the Otsu and Iterative methods are presented in Fig. 5, whose upper-left corner indicates a 100% detection rate and a 0% false alarm rate. The more the ROC curve approaches the upper-left corner, the better the test performs. Accordingly, the wavelet based  $T^2$  method, with its ROC curve closest to the upper-left corner, outperforms the two traditional methods.

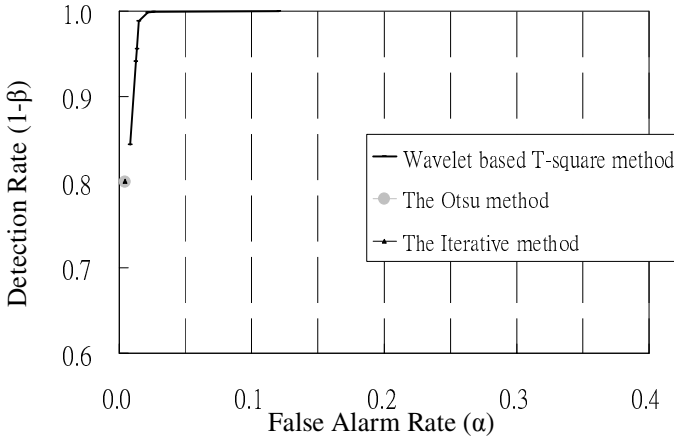


Fig. 5. The ROC plot of the Otsu, Iterative, and wavelet based  $T^2$  methods

## 4 Conclusions

This research applies the concept of the multivariate statistic  $T^2$  in quality control techniques to detect water-drop blemishes on the surfaces of LED chips. The proposed approach uses the  $T^2$  statistic values to judge the existence of water-drop blemishes through multivariate processes of combining image characteristics from wavelet decomposition of local image blocks. Experimental results show that the wavelet based  $T^2$  approach achieves an above 95% detection rate and a below 1.5% false alarm rate in detecting water-drop blemishes. As indicated in the ROC curve analysis, the wavelet based  $T^2$  method has lower false alarm rates and better detection rates than do either of the Otsu and the Iterative methods. Regarding the directions for future research opportunities, the proposed approach can be extended to detection of semi-opaque and low-contrast image defects, such as abnormal region inspection in medical images, oxidization defect detection of electronic components, and so on.

**Acknowledgments.** This study was partially supported by the National Science Council of Taiwan (R.O.C.), Project No. NSC 95-2221-E-324-034-MY2.

## References

1. Siew, L.H., Hodgson, R.M., Wee, L.K.: Texture Measures for Carpet Wear Assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10 (1988) 92-150
2. Latif-Amet, A., Ertüzün, A., Ercil, A.: An Efficient Method for Texture Defect Detection: Sub-band Domain Co-occurrence Matrices. *Image and Vision Computing*, 18 (2000) 543-553
3. Tsai, D.M., Hsiao, B.: Automatic Surface Inspection Using Wavelet Reconstruction. *Pattern Recognition*, 34 (2001) 1285-1305
4. Tsai, D.M., Wu, S.K.: Automated Surface Inspection Using Gabor Filters. *International Journal of Advanced Manufacturing Technology*, 16 (2000) 474-482
5. Jiang, B.C., Wang, C.C., Liu, H.C.: Liquid Crystal Display Surface Uniformity Defect Inspection Using Analysis of Variance and Exponentially Weighted Moving Average Techniques. *International Journal of Production Research*, 43 (2005) 67-80
6. Lin, H.D., Chiu, S.W.: Computer-Aided Vision System for MURA-Type Defect Inspection in Liquid Crystal Displays. *Lecture Notes in Computer Science*, 4319 (2006) 442-452
7. Shankar, N.G., Zhong, Z.W.: A Rule-based Computing Approach for the Segmentation of Semiconductor Defects. *Microelectronics Journal*, 37(2006) 500-509
8. Shankar, N.G., Zhong, Z.W.: Defect Detection on Semiconductor Wafer Surfaces. *Microelectronic Engineering*, 77 (2005) 337-346
9. Fadzil, M.H., Ahmed, Weng, C.J.: LED Cosmetic Flaw Vision Inspection System. *Pattern Analysis & Application*, 1 (1998) 62-70
10. Arivazhagan, S., Ganesan, L.: Texture Segmentation Using Wavelet Transform. *Pattern Recognition Letters*, 24 (2003) 3197-3203
11. Bashar, M.K., Matsumoto, T., Ohnishi, N.: Wavelet Transform-based Locally Orderless Images for Texture Segmentation. *Pattern Recognition Letters*, 24 (2003) 2633-2650
12. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. 2<sup>nd</sup> edn. Prentice-Hall, Upper Saddle River, NJ (2002) 349-403
13. Hotelling, H.: *Multivariate Quality Control*. In: Eisenhart, Hastasy, Wallis (eds.): *Techniques of Statistical Analysis*. McGraw-Hill, New York (1947)
14. Lowry, C.A., Montgomery, D.C.: A Review of Multivariate Control Charts. *IIE Transactions*, 27 (1995) 800-810
15. Montgomery, D.C.: *Introduction to Statistical Quality Control*. 5<sup>th</sup> edn, John Wiley & Sons, Hoboken, NJ. (2005) 491-504
16. Otsu, N.: A threshold selection method from gray-level histogram. *IEEE Transactions on Systems, Man, Cybernetics*, 9 (1979) 62-66
17. Jain, R., Kasturi R., Schunck, B.G.: *Machine Vision*. International edn, McGRAW-Hill, New York, NY. (1995) 80-83
18. Montgomery, D.C., Runger, G.C.: *Applied Statistics and Probability for Engineers*. 2<sup>th</sup> edn, John Wiley & Sons, New York, NY. (1999) 296-304

# Detection of Various Defects in TFT-LCD Polarizing Film

Sang-Wook Sohn<sup>1</sup>, Dae-Young Lee<sup>2</sup>, Hun Choi<sup>3</sup>,  
Jae-Won Suh<sup>4</sup>, and Hyeon-Deok Bae<sup>1</sup>

<sup>1</sup> Dep. of Electrical Engineering,  
Chungbuk National University, Korea  
{sohn6523,hdbae}@chungbuk.ac.kr

<sup>2</sup> RIUBIT, Korea  
dylee@chungbuk.ac.kr

<sup>3</sup> KRISS, Korea

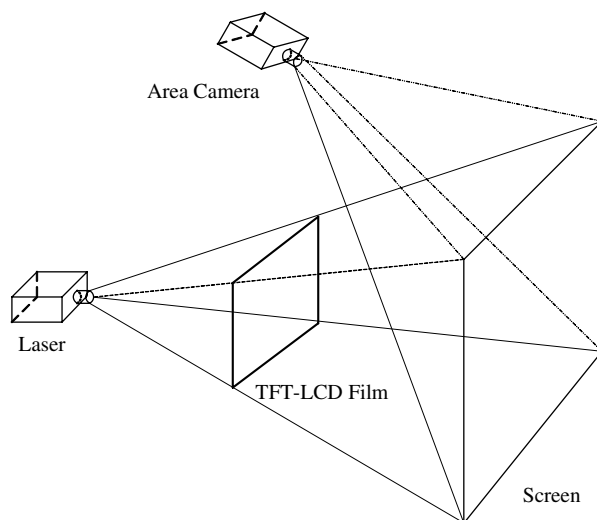
eliga@chungbuk.ac.kr

<sup>4</sup> Dep. of Electronic Engineering,  
Chungbuk National University, Korea  
sjwon@chungbuk.ac.kr

**Abstract.** The increasing use of TFT-LCDs has generated a great deal of interest in manufacturing defects on TFT-LCD polarizing film because the poor quality of TFT-LCD polarizing film result in undesirable effects on the TFT-LCD display devices. In this paper, we propose a new inspection method that reliably detects various defects of TFT-LCD polarizing films. First, we apply a least mean squares adaptive filtering technique to remove background noise. Next, we use statistical characteristics to detect possible defects. Finally, we make a binary image to identify whether the TFT-LCD polarizing film has defects or not based on an adaptive threshold value. The performance of the proposed method has been evaluated on real TFT-LCD polarizing film samples.

## 1 Introduction

As the FPD(Flat Panel Display) market becomes larger, we have increasing concerns about TFT-LCDs(Thin Polarizing film Transistor Liquid Crystal Display) because of their high visual quality requirements. The visual quality of the TFT-LCD depends on the TFT-LCD polarizing film because various defects of TFT-LCD polarizing film can generate undesirable effects on display devices. Therefore, we need to detect defects which inevitably occur during the manufacturing of the TFT-LCD polarizing film. Until recently, we have used only visual inspection to detect various defects[1]-[3]. The defects, in most cases, are not easily identified so that those persons identifying defects in a manufacturing environment have needed substantial skill and experience based on related knowledge. This human visual inspection system has some drawbacks such as inconsistency and the limitations of human sensitivity to certain defects. This system also tends to be costly. An automatic inspection system using machine vision techniques can overcome these problems.



**Fig. 1.** A machine vision system

Generally, a machine vision system suffers from non-uniformity of light and background noise. Some previous works have been studied [4]-[9] to solve the non-uniformity of light and they proposed specific algorithms based on region growing method suitable for their own system. However, these efforts have some problems: high computational complexity and long processing time. It is also true that they needed a specific algorithm to detect each defect. To overcome these problems, we suggest a new algorithm based on adaptive filtering and statistical characteristics.

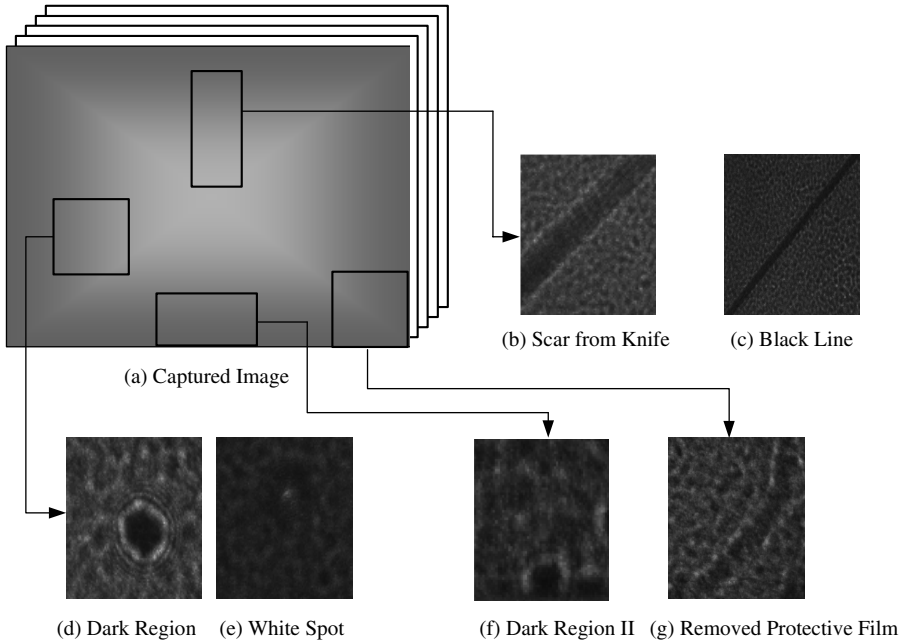
In this paper, we propose a new detection algorithm which is free from non-uniformity of light and background noise. In Sect. 2, we summarize image and the various types of defects. In Sect. 3, we discuss the detection algorithm for TFT-LCD polarizing film. To detect various defects, we apply LMS(Least Mean Square) adaptive filtering, statistical characteristics, and variable thresholding. Section 4 explains the simulation results. We draw conclusions in Sect. 5.

## 2 A Machine Vision System

### 2.1 Image Acquisition

To detect the various defects on TFT-LCD polarizing film, we use digital image processing techniques. Fig. 1 shows the concept of our machine vision system used to get a TFT-LCD polarizing film image. First, a laser source penetrates the TFT-LCD polarizing film. Then, we capture the screen image using an area camera. The proposed machine vision system has some problems: background





**Fig. 2.** Various defects on TFT-LCD polarizing film

noise and the non-uniformity of laser source. Because defects and the background have a similar gray level, it is difficult to distinguish defects from the background. In addition, the non-uniformity of the laser source and the low contrast of the defect region make it nearly impossible to apply simple thresholding method directly.

## 2.2 Various Defects on TFT-LCD Polarizing Film

Figure 2 shows various defects of the TFT-LCD polarizing film obtained by using our machine vision system. Figure 2 (a) shows the effect of the non-uniformity of the laser source. The center area is brighter than the border area. Figure (b) through (g) show the various defects: a scar from a knife, a black line, a dark region, a white spot, the dark region II, and the removed protective polarizing film, respectively. These kind of defects such as coating a protective polarizing film, rolling up a TFT-LCD polarizing film, or cutting the polarizing film occur during the manufacturing process. Figure (b) through (e) are comparatively easy to detect. However the defects shown in (f) and (g) are hard to detect because they exhibit similar gray levels between the defects and the background as shown in Fig. 3.

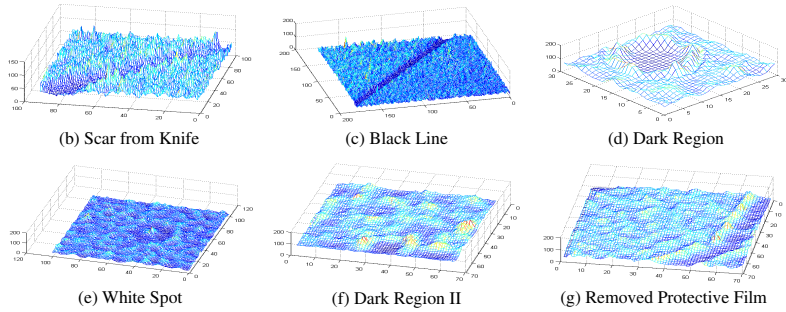


Fig. 3. 3D mesh analysis

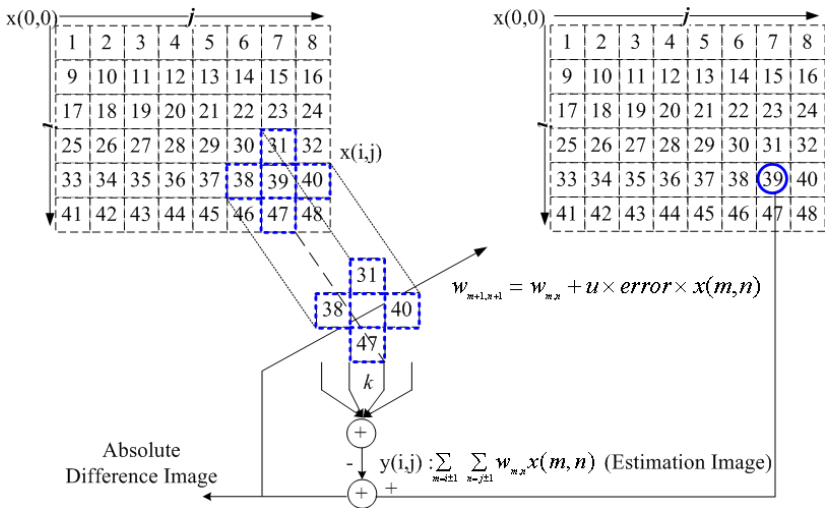


Fig. 4. Adaptive filtering

### 3 The Detection Algorithm

We propose a new detection algorithm that consists of 3 steps: LMS adaptive filtering, statistical characteristics, and variable threshold. First, we use a LMS adaptive filtering technique to remove the background noise. Then, we obtain the absolute difference image by subtracting the background image from the originally acquired image. Next, we use local statistical characteristics of the absolute difference image and variable threshold binary method to judge the defects.

#### 3.1 LMS Adaptive Filtering

The image can vary with quite different values even for the same polarizing film. This is because the contrast is affected by the statistical and physical charac-

teristics of the polarizing film and conditions in the acquired image which is called background noise. We use the LMS adaptive filter technique to remove the background noise. The use of an LMS adaptive filter offers an attractive solution to the problems as it usually provides a significant improvement in performance over the use of a fixed filter designed by conventional methods. The LMS adaptive algorithm makes use of some neighboring pixels to predict the target pixel.

Thus the LMS adaptive algorithm has encountered an error between the prediction image and target image. Most LMS algorithms have been studied in the context of reducing an error. Although, in this paper, we use the decline of prediction ability against the normal prediction technique to control the step size. After all, the LMS adaptive algorithm uses a lower step size. Then the background noise predicts well, the other side it has decline of prediction in brighter or darker than background. Thus, the error image includes the defect region with removed background noise.

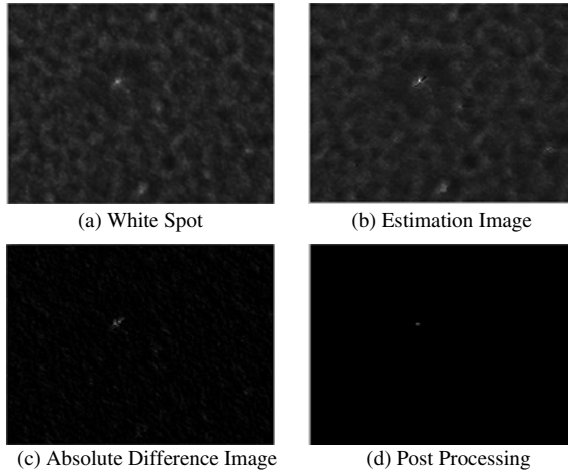
First, we estimate the background image using LMS adaptive filtering. Because it requires high computational complexity in the image processing, we just use one horizontally and vertically neighboring pixel [10], [11] as shown in Fig. 4. These surrounding four pixels have been used for every iteration to update four coefficients.

$$\begin{aligned} error &= x(i, j) - y(i, j) \\ &= x(i, j) - \sum_{m=i\pm 1} \sum_{n=j\pm 1} w(m, n) \times x(m, n) \end{aligned} \quad (1)$$

where  $x$  is original image,  $i$  and  $j$  are x-index and y-index respectively and  $w_{m,n}$  is a weighting factor.

$$w_{k+1}(m+1, n+1) = w_k(m, n) + \mu \times error \times x(m, n), \quad (2)$$

where  $\mu$  is the step size of the adaptive filter and  $k$  is iteration number. The ability of prediction depends on step size. With a high step size, the LMS adaptive filter diverges; setting a lower step size then increases the error. Accordingly, there is a need to make a good choice of adequate step size. Next, we calculate the absolute difference image between the estimated background image and the originally acquired image,  $a(i, j) = |error|$ . This acquired image is nearly free from background noise. Figure 5 shows the result of LMS adaptive filtering about white spot defect. (b) is an estimated image using adaptive filtering, and (c) is the absolute difference image between (a) and (b). In this case, we can detect the white spot defect more clearly by applying median filter to (c) as shown in (d). In addition, we can expect to obtain good results for some other defects: the scar from a knife and the black line. However, we need additional processing to detect the other defects such as the dark region II and the removed protective polarizing film.



**Fig. 5.** The processing of adaptive filtering

### 3.2 Statistical Characteristics

Although we can obtain the absolute difference image which is nearly free from background noise using LMS adaptive filtering, there are still remaining mottles in the background. To enhance the defects on the absolute difference image, we analyze the local statistical characteristics around the defects. First, we calculate the local variance

$$\delta^2 = \sum_{i=1}^m \sum_{j=1}^m (a(i, j) - LM)^2, \tag{3}$$

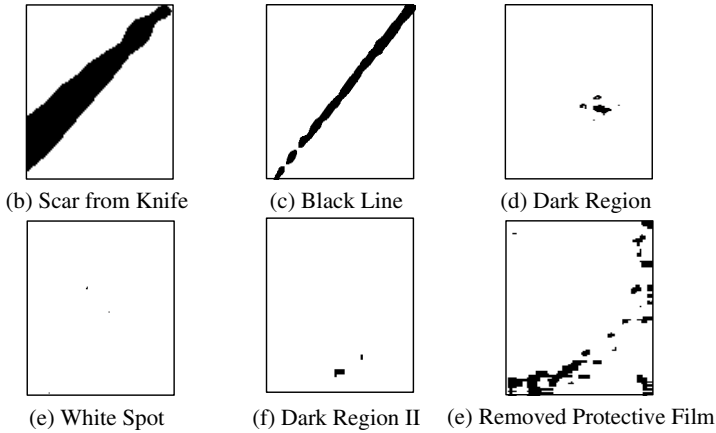
where  $LM$  is the local mean of the  $m \times m$  window. From the experimental results, we obtain the results

$$\delta_{DR}^2 < \delta_{BG}^2 < \delta_{WS}^2, \tag{4}$$

where  $\delta_{DR}^2$ ,  $\delta_{BG}^2$ , and  $\delta_{WS}^2$  is the local variance of the dark region, background noise, and white spot, respectively. DR includes dark defects like as the scar form a knife, a black line, a dark region, a dark region II and the removed protective film. WS includes white defects like as white spot and dark region. The dark region can be considered as DR or WS as its reversion. Figure 3 (d) and (e) proves this analysis. Therefore, we can distinguish the defect and background noises.

Next, we try to increase the dynamic range of the chosen features so that these features can be detected easily. The greatest difficulty in image enhancement is quantifying the criterion for enhancement. In our method, each pixel of the absolute difference image is replaced by a

$$e(i, j) = a(i, j) + \alpha \times GM, \tag{5}$$



**Fig. 6.** The simulation result of various defects

where  $\alpha$  notes the weighting factor and  $GM$  is the global mean of the absolute difference image. The  $\alpha$  value can be determined as,

$$\begin{aligned}
 \text{if } \delta^2 < \delta_{DR}^2, & \quad \text{then } \alpha = -0.2 \\
 \text{if } \delta_{DR}^2 < \delta^2 < \delta_{WS}^2, & \quad \text{then } \alpha = 0.1 \\
 \text{if } \delta^2 < \delta_{WS}^2, & \quad \text{then } \alpha = 0.5
 \end{aligned} \tag{6}$$

### 3.3 Adaptive Threshold

To determine whether the TFT-LCD polarizing film has a defect or not, we have to make a binary image using the enhanced image. To make a binary image, we must decide the threshold value. However, it is nearly impossible to fix the threshold value for the entire TFT-LCD polarizing film due to non-uniformity of the laser source [12]. Although the effect of the non-uniformity of the laser source is slightly removed by the adaptive filter technique, we have to consider the remaining non-uniformity property. Therefore, we adaptively select the variable threshold value [13] based on the ratio of the local mean.

$$Th_i = \frac{LM_i}{LM_{i-1}} \times Th_{i-1}, \tag{7}$$

where  $Th_i$  is a local threshold value and  $i$  means the index of pixel. The initial value of  $Th$  and  $LM$  is defined as a positive integer from experimental results. The overall detection procedures can be summarized as following. Step 1, we use the adaptive filtering technique to remove the background noise using equation 1 and 2. Step 2, we analyze the local statistical characteristics to enhance the defects on the absolute difference image using equation 3 through 6. Finally, we have to make a binary image using the enhanced image and equation 7.

**Table 1.** The result of defect detection for a local region of TFT-LCD polarizing film, where the BG(Background), the Kn(Scar from Knife), the BL(Black Line), the DR(Dark Region), the WS(Whit Spot), the DRII(Dark Region II), and the RP(Removed Protective polarizing film)

Num.	Def. or not	Type	Num.	Def. or not	Type	Num.	Def. or not	Type
1	Detection	DR	19	non	BG	37	Detection	RP
2	Detection	BL	20	Detection	DR	38	non	BG
3	non	BG	21	Detection	WS	39	Detection	WS
4	Detection	BL	22	Detection	RP	40	Detection	DRII
5	non	BG	23	Detection	DR	41	Detection	WS
6	non	BG	24	non	BG	42	non	BG
7	Detection	WS	25	Detection	BL	43	Detection	KN
8	non	BG	26	non	BG	44	Detection	DR
9	Detection	DRII	27	non	BG	45	Detection	BL
10	non	BG	28	non	BG	46	Detection	WS
11	Detection	RP	29	Detection	RP	47	Detection	BL
12	Detection	DRII	30	non	BG	48	Detection	DR
13	Detection	DR	31	Detection	DR	49	non	BG
14	Detection	BL	32	non	BG	50	Detection	DR
15	non	BG	33	non	BG	51	Detection	DRII
16	Detection	KN	34	non	BG	52	Detection	RP
17	non	BG	35	Detection	WS	53	Detection	WS
18	non	BG	36	Detection	DR	54	Detection	DRII

## 4 Experiments

To evaluate the performance of the proposed algorithm, we performed experiments from two points of view. In the first experiment, we verified the detection ability of the proposed algorithm with given information about which film had a defect. To make a binary image, the parameters were set the step size of adaptive filter was 0.0001, the size of the local variance window was 5 by 5, and the size of the threshold window size was 3 by 3. Figure 6 shows the experimental results we see that the proposed algorithm was successful for separation of detect and background. Second, the proposed algorithm distinguished the defects from background noises without any given information. For the second experiments, we randomly made local images with 200 by 200 pixels from the full size image. The number of sample images was 28. These local images included various defect types (2 sample images including a scar from a knife, 6 sample images including a black line, 9 sample images including a dark region, 7 sample images including a white spot, 5 sample images including the dark region II, and 5 sample images including a removed protective polarizing film). 14 sample images experimental results are shown in Table 1. The distinction abilities of the proposed algorithm were evaluated for the full size images. The total number of full size images was 131. The experimental results are shown in Table 2. From these results, we conclude that the proposed algorithm can be used for distinction of a defect and

**Table 2.** The result of defect detection for a whole region of TFT-LCD polarizing film

Defect type	Total Number	Detection	Accuracy(%)
Scar from Knife	9	9	100
Black Line	9	9	100
White Spot	16	16	100
Dark Region	54	52	96
Dark Region II	28	22	78
Removed Protective Polarizing film	15	11	73

background in the local image and full size image. However, some defects, such as the dark region II and removed protective polarizing film cannot be exactly detected.

## 5 Conclusion

For detection of various defects in TFT-LCD polarizing film, we proposed a new technique based on adaptive filtering and statistical characteristics. To acquire the image, we constructed a machine vision system using penetration property. This newly proposed machine vision system had some drawbacks: the non-uniformity of the laser source and background noise. Non-uniformity of the laser source affected the central region of TFT-LCD polarizing film as it was brighter than the border region. Because defects and background noise had a similar gray level, it was very difficult to detect the defects. To solve these problems, we used statistical characteristics based on an adaptive filter technique and variable threshold value. From the experiments performed on 131 real TFT-LCD polarizing film samples, we can conclude from the results that the detection ratio of the scar from a knife, the black line, and the white spot was 100%. However, we had to make an effort to detect the dark region and the removed protective polarizing film because their detection ratio was somewhat low. We thus expect that our proposed algorithm can be used to control the TFT-LCD polarizing film quality level.

**Acknowledgments.** This work was supported in part by grant CN-04-1-013 from the program entitled Training of Graduate Students in Regional Innovation that was conducted by the Ministry of Commerce Industry and Energy of the Korean Government.

## References

- [1] Hewlett-Packard Development Company. Understanding pixel defects in TFT-LCD flat panel monitors. Hewlett-Packard (2004)
- [2] Tannas. L. E. Jr.: Evolution of flat-panel displays. Proceedings of the IEEE **82**(4) (1994) 499–509

- [3] Definition of measurement index for luminance mura in FPD image quality inspection. SEMI standard: SEMI D31-1002. <http://www.semi.org>
- [4] Yanowitz, S. D., Brucjstein, A. M.: A new method for image segmentation. *Comput. Vision Graph. Image Process.* **46** (1989) 82–95
- [5] Pratt, W. K., Sawkar, S. S., O'Reilly, K.: Automatic blemish detection in liquid crystal flat panel displays. *IS/T/SPIE Symposium on Electronic Imaging: Science and Technology.* (1998)
- [6] Jeucke, L., Knaak, M., Orglmester, R.: A new image segmentation method on human brightness perception and foveal adaptation. *IEEE Signal Processing Letters.* **7**(6) (2000) 129–131
- [7] Kim, J.H., Ahn, S., Jeon, J. W., Byun, J.E.: A high-speed high-resolution vision system for the inspection of TFT-LCD. *Proceedings. ISIE 2001. IEEE International Symposium* **1** (2001) 101–105
- [8] Belsley, D.A., Kuh, E., Welsch, R.E.: *Regression Diagnostics*, John Wiley (1998)
- [9] Heucke, L., Knaak, M., Zhu, H.: A new image segmentation method based on human brightness perception and foveal adaptation. *IEEE Signal Processing Letters.* **7**(3) (1998) 468–473
- [10] Simon Haykin. *Adaptive Filter Theory*. Prentice Hall (2002)
- [11] Arthur R. W., Jr.: *Fundamentals of Electronic Image Processing*. SPIE Press (1996)
- [12] Otsu, N.: A threshold selection method from gray-level historam. *IEEE Trans. Syst., Man, Cybern., SMC-9* (1979) 62–66
- [13] Chen, F. H. Y., Lam, F. K., Zhu, H.: Adaptive thresholding by variational method. *IEEE Trans. Image Processing.* **7**(3) (1998) 468–473



# Dimensionality Problem in the Visualization of Correlation-Based Data

Gintautas Dzemyda<sup>1,2</sup> and Olga Kurasova<sup>1,2</sup>

<sup>1</sup> Institute of Mathematics and Informatics, Akademijos St. 4, LT 08663, Vilnius

<sup>2</sup> Vilnius Pedagogical University, Studentu St. 39, LT 08106, Vilnius, Lithuania  
{Dzemyda, Kurasova}@ktl.mii.lt

**Abstract.** A method for visualization the correlation-based data has been investigated. The advantage of this method lies in the possibility to restore the system of multidimensional vectors describing parameters from their correlation matrix (one vector for one parameter) and to visualise these vectors for the visual decision making on the similarity of the parameters. The goal of this research is to investigate the possibility to reduce the dimensionality of the vectors from the restored system and to evaluate the visualization quality in dependence on the reduction level.

## 1 Introduction

Any set of objects (cases, vectors) may often be characterized by common parameters (variables, features). A combination of values of all the parameters characterises a concrete object from the whole data set. The values obtained by any parameter can depend on the values of the other parameters, i.e. the parameters can be correlated. The problem is to discover knowledge about the interlocation of parameters, and about groups (clusters) of parameters by the values of elements of the correlation matrix.

A lot of references to real correlation matrices may be found (see [4]). Recent research and technology development applications produce correlation matrices and discover knowledge via their analysis, too: the references cover air pollution, vegetation of coastal dunes, groundwater chemistry, minimum temperature trends, zoobenthic species-environmental relationships, development and analysis of large environmental and taxonomic databases, curricula of studies (see [5], [6], [8], [9], [15]).

One of the most popular methods of analysing multidimensional data and reducing their dimensionality is the principal component analysis (PCA) [10], but it does not show an interlocation of variables – only their location around the zero-correlation. It means that we need more sophisticated means for the analysis of correlations. An attempt to visualize the correlations by using the topographic maps is made in [2]. However, only the method, proposed in [4], gives a theoretically grounded possibility for a new view to the analysis of correlations, in particular, to the visualization of data stored in correlation matrices. It provides

a multidimensional vector describing each parameter for further visualising or clustering of the set of these vectors. The method consists of two stages: restoring a system of vectors based on the correlation matrix and its visualization. A visual presentation of data stored in the correlation matrix makes it possible to discover additional knowledge hidden in the matrices and to make proper decisions.

The goal of this research is to investigate the possibility to reduce the dimensionality of the vectors from the restored system and to evaluate the visualization quality in dependence on the reduction level.

## 2 The Method of Visual Analysis of Correlation Matrices

A method for the visualization of a set of parameters (variables, features) characterised by their correlation matrix has been proposed in [4]. The method consists of two stages: building and visualization of a system of vectors based on the correlation matrix. The advantage of the method lies in the possibility to restore a system of multidimensional vectors describing variables from the correlation matrix – one vector for one variable. We present here essential items that will make the reading of the whole paper easier. Also, specific features of realizations of the method are pointed.

Denote parameters (variables, features) that characterize any object by  $x_1, \dots, x_n$ . The term “object” may cover, e.g. people, equipment, or products of manufacturing. There exist groups (clusters) of parameters that characterize different properties of the object. The correlation matrix  $R = \{r_{x_i x_j}, i, j = \overline{1, n}\}$  of parameters may be calculated by analysing the objects that compose the set. Here  $r_{x_i x_j}$  is a correlation coefficient of parameters  $x_i$  and  $x_j$ . The specific character of the correlation matrix analysis problem lies in the fact that the parameters  $x_i$  and  $x_j$  are related more strongly if the absolute value of the correlation coefficient  $|r_{x_i x_j}|$  is higher, and less strongly if the value of  $|r_{x_i x_j}|$  is lower. The minimal relationship between the parameters is equal to 0. The maximal relationship is equal to 1 or  $-1$ .

Let  $S^n$  be a subset of an  $n$ -dimensional Euclidean space  $R^n$  containing vectors of unit length, i.e.  $S^n$  is a unit sphere,  $\|Y\| = 1$  if  $Y \in S^n$ . It is necessary to determine a system of vectors  $Y_1, \dots, Y_n \in S^n$  corresponding to the system of parameters  $x_1, \dots, x_n$  so that  $\cos(Y_i, Y_j) = |r_{x_i x_j}|$  or  $\cos(Y_i, Y_j) = r_{x_i x_j}^2$ . If only the matrix of cosines  $K = \{\cos(Y_i, Y_j), i, j = \overline{1, n}\}$  is known, it is possible to restore the system of vectors  $Y_s = (y_{s1}, \dots, y_{sn}) \in S^n, s = \overline{1, n}$ , as follows:

$$y_{sk} = \sqrt{\lambda_k} \alpha_{sk}, \quad k = \overline{1, n}, \tag{1}$$

where  $\lambda_k$  is the  $k$ -th eigenvalue of the matrix  $K$ ,  $\alpha_k = (\alpha_{1k}, \dots, \alpha_{nk})$  is a normalized eigenvector corresponding to the eigenvalue  $\lambda_k$ . We see from (1) that the  $k$ -th element  $y_{sk}$  of vector  $Y_s$  depends on the  $k$ -th eigenvalue  $\lambda_k$  of the matrix  $K$  and on the  $s$ -th component  $\alpha_{sk}$  of the eigenvector  $\alpha_k$  corresponding to  $\lambda_k$ . The system of vectors  $Y_1, \dots, Y_n \in S^n$  exists, if the matrix of their scalar products is non-negative definite. The correlation matrix  $R = \{r_{x_i x_j}, i, j = \overline{1, n}\}$

is non-negative definite. It has been proven in [4] that the matrix  $R^2 = \{r_{x_i x_j}^2, i, j = \overline{1, n}\}$  is non-negative definite as well. Therefore, the system of vectors  $Y_1, \dots, Y_n \in S^n$  may be restored when  $R$  or  $R^2$  are used as  $K$ .

The set of vectors  $Y_1, \dots, Y_n \in S^n$ , which corresponds to the set of parameters  $x_1, \dots, x_n$ , may be mapped on a plane trying to preserve a relative distance between  $Y_1, \dots, Y_n \in S^n$ . This leads to the possible visual observation of a layout of parameters  $x_1, \dots, x_n$  on the plane.

There exist lots of methods that can be used for reducing the dimensionality of data by projecting high-dimensional data sets as points on a low-dimensional, usually 2D, display. A detailed review of the methods is given in [12]. We apply here two methods: Sammon's mapping, that belongs to the so-called metric multidimensional scaling methods, and the self-organizing map (SOM) that is a neural network. These two visualization methods are based on different principles, and, therefore, they supplement each other when used jointly. Some (sometimes sufficient) knowledge on a set of parameters may be obtained by using individual methods. On most cases, however, the necessity and efficiency of their join use is unquestionable: this allows us to observe the same data set from various standpoints and extend our knowledge on the object of investigation.

The analysis of relative performance of the different algorithms in reducing the dimensionality of multidimensional vectors [1] indicates Sammon's mapping [14] to be still one of the best methods of this class. The direct application of Sammon's method allows to reduce the dimensionality of vectors  $Y_1, \dots, Y_n \in S^n$  by computing a correspondent system of two-dimensional vectors  $Z_1, \dots, Z_n \in R^2$ , where  $Z_s = (z_{s1}, z_{s2})$ ,  $s = \overline{1, n}$ .

The self-organizing map (SOM), proposed by Kohonen [12], is a class of neural networks that are trained in an unsupervised manner using competitive learning. It is a well-known method for mapping a high dimensional space onto a low dimensional one (usually two-dimensional grid). The method allows putting complex data into order based on their similarity and shows a map from which the features of the data can be identified and evaluated. A variety of realizations of the SOM have been developed (see e.g. [11], [13]). All of them produce different results to some extent. The realization below is similar to that proposed and examined in [4]. The advantages of the SOM are its unsupervised training and that it combines clustering and projection operations. The last advantage allows increasing the quality of visualization by additional application of Sammon's mapping to the neurons of the SOM: here we get some combined mapping.

Usually, the neurons are connected to each other via a rectangular or hexagonal topology. In this paper, we consider the rectangular case, only. The rectangular SOM is a two-dimensional array of neurons  $M = \{m_{ij}, i = \overline{1, k_x}, j = \overline{1, k_y}\}$ . Here  $k_x$  is the number of rows, and  $k_y$  is the number of columns. The dimension of the vectors, which will be presented as inputs to train the network, is  $n$ . Each component of the input vector is connected to every individual neuron. Thus, there is a connection between the neuron of the network and every component of the input vector. The weights of these connections form an  $n$ -dimensional synaptic weight vector (the codebook vector). Thus, any

neuron is entirely defined by its location on the grid (the number of row  $i$  and column  $j$ ) and by the codebook vector, i.e. we can consider a neuron as an  $n$ -dimensional vector  $m_{ij} = (m_{ij}^1, m_{ij}^2, \dots, m_{ij}^n)$ . In this way, each vector (neuron)  $m_{ij}$  represents a part of  $S^n$  because  $Y_1, \dots, Y_n \in S^n$ , but, in most cases, the vector  $m_{ij}$  itself does not belong to  $S^n$ , i.e.  $m_{ij} \notin S^n$ .

The SOM learning starts from the vectors  $m_{ij}$  initialised randomly. It is proposed in [4] to select the starting values of  $m_{ij}$  so that cosines between their pairs be positive, just like between the pairs of vectors from the training set  $\{Y_1, \dots, Y_n\}$ . Here we will follow [4].

At each learning step, an input vector  $Y$  is drawn from the training set  $\{Y_1, \dots, Y_n\}$  and passed to the neural network. A learning iteration consists of  $n$  learning steps: the input vectors from  $Y_1$  to  $Y_n$  are passed to the neural network in consecutive or random order. The consecutive order was used in [4]. In this paper, we use the random order, because we try to eliminate the influence of numeration of the input vectors on the learning process. The whole learning process consists of  $v$  iterations. Like in [4],  $v = 200$  was set in our experiments.

Using the SOM-based approach above, we can draw a table with cells corresponding to the neurons. The cells corresponding to the neurons-winners are filled with the order numbers of vectors  $Y_1, \dots, Y_n$ . Some cells may remain empty (see Table 1 for example, comments on data in the table are presented in Section 4). One can make a decision visually on the distribution of the vectors  $Y_1, \dots, Y_n$  in the  $n$ -dimensional space in accordance with their distribution among the cells of the table. However, the table does not answer the question, how much the vectors of the neighboring cells are close in the  $n$ -dimensional space.

Several ways of integration of the Sammon's mapping and the SOM are possible. All they are based on the fact that the input data and results of the SOM training are  $n$ -dimensional vectors, and the Sammon's mapping, like any other metric multidimensional scaling method, performs the projection of  $n$ -dimensional vectors onto the lower dimensionality (e.g. plane). In the SOM, several elements (neurons) of the two-dimensional rectangular grid are activated (become winners), while the remaining elements are not activated. The activated elements of the grid may be considered as points on the plane. The number of row and column characterises any of these elements, i.e. the location of these elements is fixed on the plane by the nodes of the rectangular grid. The elements are characterized by  $n$ -dimensional vectors, too. A natural idea comes to apply the distance-preserving projection method to additional mapping of vectors-winners in the SOM. Sammon's mapping may be used for such purposes. Such a combination of mapping methods is examined and grounded experimentally in [4]. This is a consecutive way of integration. More sophisticated way is proposed in [7]. Here the multidimensional vectors are analysed taking into account the learning flow of the SOM. The quality of this way is better as compared with the consecutive way. Therefore, the last way of integration is used in the further experiments.

### 3 Data Set

The experiments were carried out on the basis of the correlation matrix with the known “ideal” partition of parameters into groups. The known “ideal” partition serves as a basis for evaluating the results of analysis. The experiment was carried out using the correlation matrix  $R_{24}$  of 24 psychological tests on 145 pupils of the 7th and 8th forms in Chicago (see [3], [4]). The correlation matrix is given in [4]. There are five groups of tests:

- 1) spatial perception  $\{x_1, \dots, x_4\}$ ,
- 2) verbal tests  $\{x_5, \dots, x_9\}$ ,
- 3) the rapidity of thinking  $\{x_{10}, \dots, x_{13}\}$ ,
- 4) memory  $\{x_{14}, \dots, x_{19}\}$ ,
- 5) mathematical capabilities  $\{x_{20}, \dots, x_{24}\}$ .

The tests of the fifth group characterise a general development of the tested person. They do not characterize separate parts of his intellect. Thus, classifying all the tests into four groups the algorithms distribute the tests of the fifth group among the other four groups. The investigations in [4] indicated that the optimal partition of parameters is as follows:  $A_1 = \{x_1, \dots, x_4, x_{20}, x_{22}, x_{23}\}$ ,  $A_2 = \{x_5, \dots, x_9\}$ ,  $A_3 = \{x_{10}, \dots, x_{13}, x_{21}, x_{24}\}$ ,  $A_4 = \{x_{14}, \dots, x_{19}\}$ . We will consider this partition as an “ideal” one.

### 4 Results of the Analysis

A set of vectors  $Y_1, \dots, Y_n$  was calculated on the basis of matrix  $R_{24}$  ( $n = 24$ ) using the approach of Section 2. Elements of matrix  $R_{24}$  are positive. Therefore, their values were not squared for analysis. Each vector  $Y_1, \dots, Y_n$  contains 24 elements. However, the smaller number of elements may be used when analysing the set of vectors  $Y_1, \dots, Y_n$ . The reason is that the values of elements depend on the size of eigenvalues  $\lambda_k$ ,  $k = \overline{1, n}$  (see (1)). Let the eigenvalues be numbered in the decreasing order of their size:  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ . Therefore, the elements of vectors  $Y_1, \dots, Y_n$ , having larger order numbers, depend on the smaller eigenvalues. This leads to idea, that it is expedient to reduce the length of the vectors by ignoring and eliminating their elements with larger order numbers. The goal of analysis is to evaluate the influence of the length of vectors  $Y_1, \dots, Y_n$  to the projection of these vectors on the plane. Two ways of projection are investigated – direct Sammons’s mapping and the integrated mapping that combines the SOM and Sammon’s mapping. The SOM of size  $3 \times 3$  was used in the experiments.

The mapping results using the SOM, when only the first two elements of vectors  $Y_1, \dots, Y_{24}$  are considered, i.e.  $Y_s = (y_{s1}, y_{s2})$ ,  $s = \overline{1, 24}$ , is presented in Table 1a. In the SOM in Table 1b, the first three elements of the vectors  $Y_1, \dots, Y_{24}$  are considered, i.e.  $Y_s = (y_{s1}, y_{s2}, y_{s3})$ ,  $s = \overline{1, 24}$ . In Tables 1c, 1d, 1e and 1f, the number of elements of the visualised vectors grows and takes values 5, 9, 14, 24 respectively. We see four definite clusters of parameters,

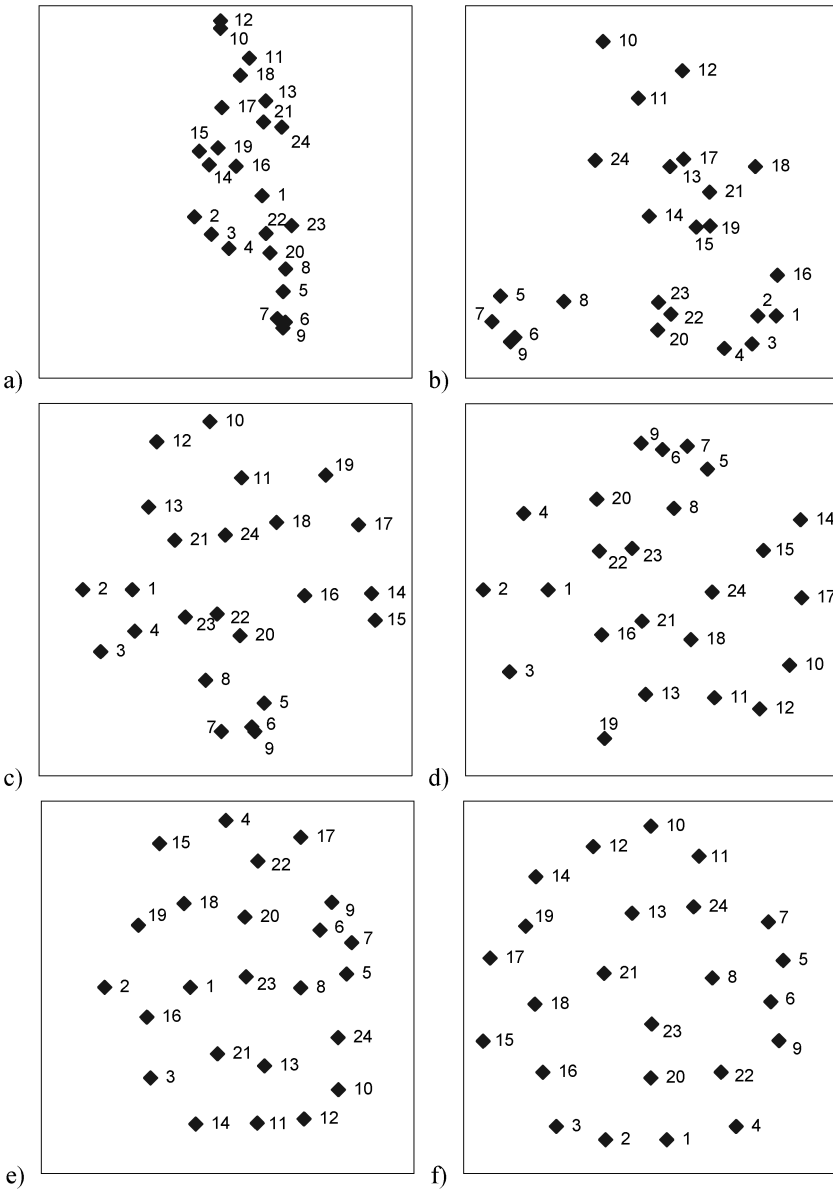
when all 24 elements are considered, i.e. when  $Y_s = (y_{s1}, \dots, y_{s24})$ ,  $s = \overline{1, 24}$  (see Table 1f). These clusters correspond to the “ideal” partition discussed in Section 3. However, results of Table 1e are also good, because parameters  $x_{20}$  and  $x_{21}$  of the fifth group, whose all five parameters  $x_{20} - x_{24}$  are assigned to other four groups in the “ideal” partition, now are placed in separate cells of the SOM. This indicates an existence of some specific character in the parameters of the fifth group.

In Figures 1 and 2, we observe the distribution of the parameters, characterised by their correlation matrix, on a plane. We do not present legends and units for both axes in the figures, because we are interested in observing the interlocation of points corresponding to the parameters on a plane only. Numbers in the figures are order numbers of parameters  $x_1, \dots, x_n$  and corresponding vectors  $Y_1, \dots, Y_n$ .

In Fig. 1, we present the distribution of vectors  $Y_1, \dots, Y_{24}$  on a plane after the direct application of Sammon’s method in dependence on the number of considered elements of these vectors: the first two, three, five, nine, fourteen and all 24 elements.

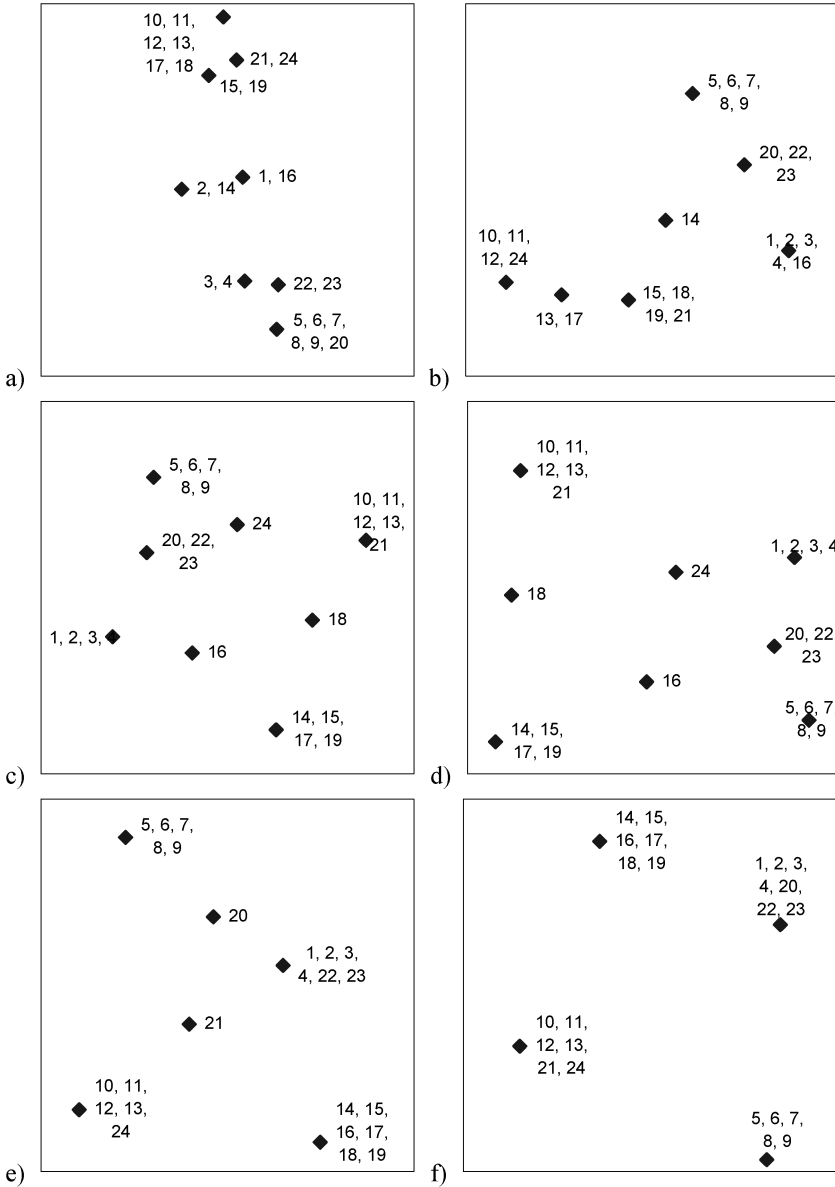
**Table 1.** 3x3 SOM, when only the first two (a), three (b), five (c) nine (d), fourteen (e) and all 24 (f) elements of the vectors are considered

a)	2, 14 15, 19 10, 11, 12, 13, 17, 18	3, 4 1, 16 21, 24	5, 6, 7, 8, 9, 20 22, 23
b)	15, 18, 19, 21 1, 2, 3, 4, 16	13, 17 14 20, 22, 23	10, 11, 12, 24 5, 6, 7, 8, 9
c)	14, 15, 17, 19 18 10, 11, 12, 13, 21	16 24	1, 2, 3, 4 20, 22, 23 5, 6, 7, 8, 9
d)	10, 11, 12, 13, 21 18 14, 15, 17, 19	24 16	5, 6, 7, 8, 9 20, 22, 23 1, 2, 3, 4
e)	10, 11, 12, 13, 24 14, 15, 16, 17, 18, 19	21	1, 2, 3, 4, 22, 23 20 5, 6, 7, 8, 9
f)	14, 15, 16, 17, 18, 19 10, 11, 12, 13, 21, 24		1, 2, 3, 4, 20, 22, 23 5, 6, 7, 8, 9



**Fig. 1.** Results of the direct application of Sammon's method, when only the first two (a), three (b), five (c) nine (d), fourteen (e) and all 24 (f) elements of the vectors are considered

As we see in Figures 1e and 1f, the parameters are almost uniformly distributed after the direct compression of high (14 and 24) dimensional points on a plane by Sammon's mapping. Nevertheless, some decision on the interlocation of parameters may be made: more similar parameters are located nearer



**Fig. 2.** Results of the integrated mapping, when only the first two (a), three (b), five (c) nine (d), fourteen (e) and all 24 (f) elements of the vectors are considered

in the figures. When the number of considered elements of vectors  $Y_1, \dots, Y_{24}$  is smaller, the certain clusters of points may be observed in two-dimensional projection (see Figures 1c and 1d). However, these clusters are far from “ideal” ones (see Table 1f and [3], [4]).



The results of integrated mapping are presented in Fig. 2. It shows the distribution of vectors  $Z_s = (z_{s1}, z_{s2})$ ,  $s = \overline{1, 24}$ , obtained after an application of Sammon's method to the vectors-winners of the SOM. The better results (the clusters are more expressed and seen visually) are obtained when analysing vectors containing the larger number of the elements (see Figures 2e and 2f).

## 5 Conclusions

A method for visualization the correlation-based data set of parameters has been examined. The advantage of this method lies the possibility to restore the system of the multidimensional vectors describing variables from the correlation matrix and to visualise these vectors for the visual decision making on the similarity of parameters. The method consists of two stages: restoring a system of vectors and its visualization. Sammon's mapping and its integrated combination with the self-organizing map were applied in the visualization. The possibility to reduce the dimensionality of the vectors from the restored system in dependence on the size of eigenvalues of the correlation matrix is investigated.

The research showed the dependence of visualization results on the length of the vectors that characterize the individual parameter.

The parameters are almost uniformly distributed after a direct compression of high-dimensional points on a plane. Nevertheless, some decision on the interlocation of parameters may be made: we can visually evaluate the interlocation of parameters. When the number of elements of visualized vectors is taken smaller, the clusters of points may be observed in two-dimensional projection, but the clusters are far from "ideal" ones. The opposite conclusion may be drawn for the SOM and the integrated mapping: the better results are obtained when analysing vectors containing the larger number of elements; the best results are obtained when the vectors of whole length are used.

The experiments prove the efficiency of the integrated mapping. It allows to observe the clusters in the data set on the correlated parameters not only when the vectors of whole length are used, but in case of their reduced length, too. Moreover, the application of smaller number of elements in the visualised vectors allowed to discover an existence of some specific character in the parameter set.

The reduction of the dimensionality of the vectors from the restored system allows to increase the quality of decisions when the direct mapping is used. In this case, the number of elements in the vectors, characterizing parameters, should be small. Its size needs a deeper investigation.

## References

1. Bezdek, J.C., Pal, N.R.: An index of topological preservation for feature extraction. *Pattern Recognition*, **28**, 381–391 (1995)
2. Díaz, I., Cuadrado, A., Diez, A.: Correlation visualization of high dimensional data using topographic maps. *Artificial Neural Networks – ICANN 2002, Lecture Notes in Computer Science*, vol. 2415, Springer (2002), 1005–1010

3. Dzemyda, G.: Clustering of parameters on the basis of correlations via simulated annealing. *Control and Cybernetics, Special Issue on Simulated Annealing Applied to Combinatorial Optimization*, **25**(1), 55–74 (1996)
4. Dzemyda, G.: Visualization of a set of parameters characterized by their correlation matrix. *Computational Statistics and Data Analysis*, **36**(1), 15–30 (2001)
5. Dzemyda G.: Visualization of the correlation-based environmental data. *Environmetrics*, **15**(8), 827–836 (2004)
6. Dzemyda, G.: Multidimensional data visualization in the statistical analysis of curricula. *Computational Statistics and Data Analysis*, **49**, 265–281 (2005)
7. Dzemyda, G., Kurasova, O.: Heuristic approach for minimizing the projection error in the integrated mapping. *European Journal of Operational Research*, **171**(3), 859–878 (2006)
8. Fautin, D.G., Buddemeier, R.W.: *Biogeoinformatics of Hexacorallia (Corals, Sea Anemones, and Their Allies): Interfacing Geospatial, Taxonomic, and Environmental Data for a Group of Marine Invertebrates* (2001), <http://www.kgs.ukans.edu/Hexacoral/Envirodata/Correlations/correl1.htm>
9. Ieno, E.N.: *Las Comunidades Bentónicas de Fondos Blandos del Norte de la Provincia de Buenos Aires: Su rol Ecológico en el Ecosistema Costero*. Tesis Doctoral Universidad Nacional de Mar del Plata Zoobenthic Species Measured in an Intertidal Area in Argentina (2000), <http://www.brodgar.com/benthos.htm>
10. Jolliffe, I.T.: *Principal Component Analysis*. Springer Verlag, Berlin (1986)
11. Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J.: *SOM\_PAK: The Self-Organizing Map Program Package*. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland (1996)
12. Kohonen, T.: *Self-Organizing Maps*, 3rd ed. Springer Series in Information Sciences, 30. Springer (2001)
13. Murtagh F.: *Fionn Murtagh's Multivariate data Analysis Software and Resources Page*, <http://newb6.u-strasbg.fr/~fmurtagh/mda-sw/>
14. Sammon, J.W.: A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, **18**, 401–409 (1969)
15. Smith, R.L.: *CBMS Course in Environmental Statistics* (2001), <http://www.stat.unc.edu/postscript/rs/envstat/env.html>

# A Segmentation Method for Digital Images Based on Cluster Analysis\*

Héctor Allende<sup>1,3</sup>, Carlos Becerra<sup>1</sup>, and Jorge Galbiati<sup>2</sup>

<sup>1</sup> Universidad Técnica Federico Santa María, Departamento de Informática,  
Casilla 110-V, Valparaíso, Chile

{hallende, cbecerra}@inf.utfsm.cl

<sup>2</sup> Pontificia Universidad Católica de Valparaíso, Instituto de Estadística,  
Casilla 4059, Valparaíso, Chile

jorge.galbiati@ucv.cl

<sup>3</sup> Universidad Adolfo Ibañez, Facultad de Ciencia y Tecnología,  
Balmaceda 1625, Viña del Mar, Chile

hallende@uai.cl

**Abstract.** We present a method for image segmentation, that is, to identify image points with an indication of the region or class they belong to. The proposed algorithm basically consists of two stages. First it starts by restoring the image from possible contamination. In the second stage it analyzes each pixel using a 3x3 sliding window. For the first pixel, it creates an objects consisting of that same pixel, and registers this object in an array. In the subsequent steps, a cluster analysis is applied to the surrounding eight pixels, an determines whether the central pixel belongs to one of the existing objects, or a new object has to be created, and registered in the array of objects.

## 1 Introduction

In the last years Digital Image Processing has been widely used by different disciplines such as: Medicine, Biology, Physics, Engineering, among others.

Among the techniques for digital image processing we find image segmentation, which comes from numerous psychological studies that show the preference of humans to group visual regions in terms of proximity, similarity and continuity, to construct sets of significant units. This approach is considered one of the most important elements within any automated system of vision, since it is the first step in the task of understanding an image and it severely affects later process in the interpretation of the image, providing useful structures for the application to the area of interest.

Regarding the definition of image segmentation, there is certain confusion, or rather, different approaches depending on the problem that is to be solved. Authors like [6], consider sufficient to identify the image points (pixels) with

---

\* This work was supported in part by Research Grants FONDECYT 1040365 and 7060040 and in part by Research Grant DGIP-UTFSM.

an indication of the region or class they belong to, while other authors such as [2] and [3] argue that it is also necessary to provide a mechanism that allows a symbolic representation of the topological relations existing between the different units. This disagreement comes mainly from different levels of abstraction associated to the segmentation process. At a low level, we only wish to divide the image in regions that have common characteristics. In this scheme the significant unit that determines the segmentation corresponds to the pixels, regions or contours that show or distinguish a similarity in terms of intensity, color, texture, among others. However, at a higher abstraction level, the idea is to generate a segmentation that identifies objects within the image, associated to a higher level of knowledge of the units contained in it. We not only want to divide the image in different regions that have common characteristics, but those regions must have a relation with objects pertaining to the dominion of the problem that is being faced. In this work it will be understood that segmentation of images corresponds to the first approach described here.

An important problem related to segmentation is to define objective measures of quality of its results, since the measures of quality of procedures such as cancer detection, mineral processing, detection of imperfections in production processes, etc., directly depends on the segmentation technique. In this work a technique for segmentation of images based on cluster analysis is proposed which, we believe, improves the quality of the segmentation, as compared with other existing techniques. See [4] and [6].

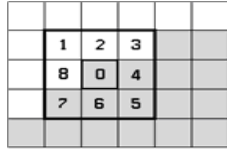
A great amount of algorithms has been developed, see [6], the problem is to select the most suitable approach for each type of problem, that is why an important aspect related to image segmentation is the selection of the algorithms to be used, to measure its performance and to understand its impact in the analysis of the image. This is the second problem analyzed in this work in which, on the one hand, a technique to evaluate and compare segmentation methods will be applied to allow selecting the appropriate segmentation algorithm to certain problem and, on the other hand, the efficiency of a new proposed technique will be validated.

## 2 Proposed Image Segmentation Algorithm

The proposed algorithm for image segmentation is based on cluster analysis to assign each pixel to an object within the image. Initially the algorithm uses a restoration filter developed in [1].

The proposed image segmentation algorithm consists basically of two main stages.

In the first stage a hierarchical agglomerative clustering algorithm is applied; the algorithm travels over the image using a 3x3 pixel sliding window, as shown in Fig. 1 the central pixel is analyzed in each window, starting with 8 conglomerates that correspond to all the pixels around it, then; in each iteration a merging of the two conglomerates that have the least "distance" occurs, a distance that will



**Fig. 1.** Sliding  $3 \times 3$  pixel window for cluster analysis. The central pixel  $Px(0)$  is being analyzed, based on the eight pixels  $Px(1)$  to  $Px(8)$ .

be defined as the smallest difference between the medium value of gray levels in each conglomerate, that is:  $D(A, B) = d(\bar{a}, \bar{b})$ .

This hierarchical agglomerative clustering algorithm in each iteration reduces the number of conglomerates until finally only one is left, however, determining the amount of conglomerates that should be has to do with the maximum distance that can be between each of them. This is defined in the following way: If in  $k - th$  iteration conglomerates A and B were grouped at a distance  $D(A, B)$ , thus the distance of the group is defined as the  $k - th$  iteration as  $D_k = D(A, B)$ . The sequence of  $D_1, D_2, \dots, D_k$  values is incremental, since in each iteration the distance among conglomerates is greater and for this reason, the maximum distance between two conglomerates must be determined, that is, if the distance in  $D_{k+1}$  is greater that a  $T_{cluster}$  threshold value, then the grouping process must finish in the  $k - th$  iteration, thus obtaining a conglomerate optimal value given by  $9 - k$ .

Once the optimum number of conglomerates is defined, the average value of the gray levels for each of them is stored, and the cluster owning the central pixel is calculated (of the sliding window of  $3 \times 3$ ) in accordance with the distance previously defined, storing this value in  $M_{current}$ .

There are two options in the second stage of the segmentation process:

a) The first sliding window within the image is analyzed, since this is a special case in the process, where the first object found in the image is created. In this case, the central pixel is assigned with the gray levels average value of the cluster to which belongs, called  $\bar{v}$ . Then, an array called  $V_{means}$  is created and the value of  $M_{current}$  is stored in its first position.

In this arrangement the objects found throughout the process of segmentation will be stored so, every time an average is added, a new object will be added as well.

b) When it is not the first time that a sliding window is analyzed we use the indicator  $T_{means}$  to see the value to be assigned to the studied pixel, lets recall that in the first stage of the algorithm it was determined the possible cluster that could own the central pixel and this value was stored in  $M_{current}$ .

At this stage it is decided whether the pixel examined belongs to an existing object or is part of a new object that must be created. To make this decision, the next steps must be followed:

1. Verify if the following condition is met for any of the objects stored in  $V_{\text{mean}}$ :

$$|V_{\text{means}}(i) - M_{\text{current}}| \leq T_{\text{means}}, \quad i = 1, 2, \dots, N, \quad (1)$$

where  $N$  is the amount of objects found up to the moment.

If condition (1) is met in at least one of the averages stored in  $V_{\text{means}}$ , it means that the studied pixel belongs to any of the objects, so then we look for the object that owns the pixel. To do this, we search again in  $V_{\text{means}}$  and we store in  $M_{\text{current}}$  the average stored in  $V_{\text{means}}$  that has the lowest value for the expression

$$|V_{\text{means}}(i) - M_{\text{current}}|, \quad i = 1, 2, \dots, N, \quad (2)$$

where  $N$  is the amount of objects found to the moment. Once it has been decided which object owns the studied pixel, the object is assigned the value of  $M$ .

If condition (1) is not met, the value of  $M_{\text{current}}$  is added as a new component of  $V_{\text{means}}$  and the value of  $M_{\text{current}}$  is assigned to the studied pixel.

These two stages are applied to every studied pixel, until all pixels have been analyzed and, therefore, the image has been segmented.

The value for  $T_{\text{cluster}}$  will be 25 and was proposed by [1], based on a series of 72 experiments on different percentages of pollution and standard deviation, besides different values of  $T_{\text{cluster}}$ . The value of  $T_{\text{means}}$  will be evaluated in Sect. 4, on the basis of the evaluation algorithm proposed in Sect. 3.

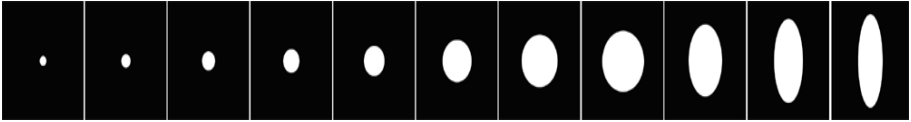
### 3 Evaluation and Comparison Method of Segmentation Algorithms

Nowadays in the field of image segmentation there is little objectivity at the moment of evaluating the different segmentation algorithms. In the works of [6], it is declared that: A human being is the best method to evaluate the result of a segmentation algorithm, but its disadvantage lies in the fact that it is a subjective method. To become an alternative to this proposal, other methods to compare and evaluate image segmentation algorithms have been developed.

Several techniques and measures of performance have been introduced to evaluate and compare the proposed algorithm with other algorithms. Some of the main phases are:

The image construction to make tests is a top priority task, see [8]. Two image subsets must be constructed for the study, the first is a subset called Shapes and the second is called Sizes, both are shown in Fig. 2, the 8 left columns belong to the Sizes subset and the 4 right columns to the Shapes subset. It must be noticed that the last column of the Sizes subset is also the first column of the Shapes subset. The evaluation and comparison experiments will be performed in later chapters based on these two internally related subsets.

To quantify the performance an algorithm developed by [7] will be applied, which proposes a supervised method that uses a reference segmentation and measures the difference between the reference segmentation and the result of



**Fig. 2.** Objects of different sizes and shapes, the diameter of the ring varies in every column. The 8 objects, from left to right, have diameters of 20, 28, 40, 50, 64, 90, 112 and 128 pixels, respectively, and they are labeled from 1 to 8. The shape also changes from a circle to an elongated ellipse.

the segmentation algorithm. Therefore, it is assumed that the reference image has three regions: the background of the region ( $B$ ) made up of pixels classified as background, an Object region ( $O$ ) made up of pixels classified as object, and an uncertain region made up of the remaining pixels. The result of the automatic segmentation algorithm has two regions: The background ( $B$ ) and the object ( $O$ ). For an image containing object and background, the probabilistic error is defined as:

$$P(error) = P(O)P(O|B) + P(B)P(O|B), \tag{3}$$

where  $P(R_1|R_2)$  is the probability of classifying  $R_2$  as  $R_1$ , and  $P(R_1)$  is the prior probability of class  $R_1$ . This measurement is often used to evaluate segmentation techniques based on thresholds. The evaluation method proposed is based on the aforementioned indicators. To calculate the difference between the reference and the segmentation algorithm, the error of "small segmentation" ( $SS$ ) is calculated first, and then the error of "over segmentation" ( $OS$ ) is calculated. The size of the object analyzed is considered to calculate these two indicators. The  $SS$  and  $OS$  errors are defined as

$$BS = Area(O|\overline{O})/A \tag{4}$$

$$BS = Area(B|\overline{B})/A, \tag{5}$$

where the operation  $\overline{\phantom{x}}$  is defined as:

$$R_1/R_2 = \{p|p \in R_1, p \notin R_2\}, \tag{6}$$

where the area ( $R$ ) is the area of region  $R$ , and  $A$  is the average of the area of objects obtained by the manual segmentation. A measure of total difference ( $TD$ ) is given by:

$$TD = BS + SS . \tag{7}$$

To obtain more information about the difference, the  $UMA$  indicator is calculated for some selected components (for instance eccentricity, area, perimeter, and so on). Then the  $UMA$  indicator is normalized by the average of the components used from the manual segmentation, that is:

$$UMA = |x - \overline{x}|/x . \tag{8}$$

The values of the different measures ( $SS$ ,  $OS$ ,  $TD$  and  $UMA$ ) are first calculated for each object and then an average for all objects is obtained. The results

of  $SS$ ,  $OS$  and  $UMA$  indicators are between 0 and 1, and are inversely proportional to the result of the segmentation. The values  $DT$  indicator can have are between 0 and 2, and the more it increases its value, the worst is the result of the segmentation.

## 4 Empirical Results

In this chapter we present a first stage of the evaluation of the proposed algorithm made through  $SS$ ,  $OS$ ,  $TD$  and  $UMA$  indicators described in Sect. 3, to see how the algorithm behaves with different parameter values, and with several test images, which are different both in shape, grey tonality and levels of noise.

In the second phase the Image Segmentation Algorithm based in Cluster Analysis (ISACA) will be compared to the:

- Classic thresholding algorithm [6].
- Region growing algorithm [6].
- Neural networks algorithm [5].

In the third phase, the results obtained will be applied to real images obtained from the database for image segmentation of Berkeley University.

### 4.1 Evaluation of the Proposed Algorithm

To evaluate the algorithm the test images defined in Sect. 3 of the shape set were used, three different grey levels contrasts were also used for each image of the shape set, the contrast level were:

- First contrast: Background=20, Object=223.
- Second contrast: Background=20, Object=102.
- Third contrast: Background=182, Object=178.

Four noise levels were also used:

- Case 1: P=1 percent, sigma=80.
- Case 2: P=7.5 percent, sigma=20.
- Case 3: P=10 percent, sigma=80.
- Case 4: P=25 percent, sigma=40.

Where  $P$  is the percentage of contaminated pixels and sigma, the standard deviation of the noise generated with a Gaussian distribution of 0 average and standard deviation  $\sigma$ .

For this procedure we took every image in the shapes set with all three grey level contrasts and all four noise levels, 20 experiments were made for each image constructed and in each the proposed algorithm, and the optimal  $T_{\text{means}}$  value was searched so as to apply a configuration that gets the best performance. The value of  $T_{\text{means}}$  was changed from 5 to 255 with an increase of 5, and for every one of them the values of  $SS$ ,  $OS$ ,  $TD$  and  $UMA$  indicators were calculated. Based on this procedure the following results were obtained:



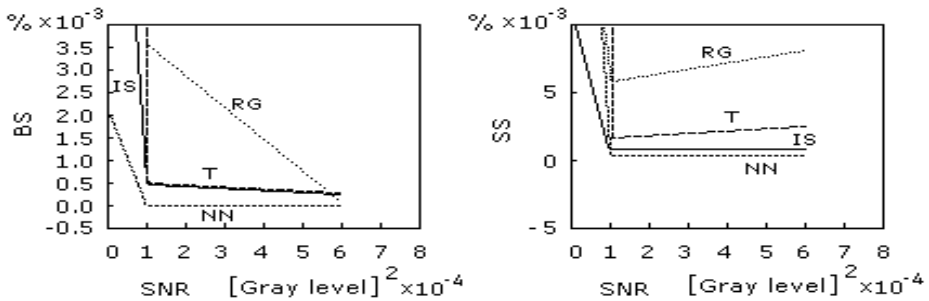
For the first contrast the best values are obtained with a 170  $T_{\text{means}}$  value, apart from the shape. For the second contrast, good results are obtained with a 70  $T_{\text{means}}$  value. For the third contrast the same is done and only desirable results are obtained from the indicators with  $T_{\text{means}}$  of 2 and 4 (in this case only extra experiments were made, changing  $T_{\text{means}}$  from 1 to 5, one by one), but as it was said before these values are higher than in the other noise levels.

### 4.2 Comparison with Other Methods

To make comparisons the mentioned algorithms will be applied. Only *SS* and *OS* indicators will be used since they give more information about total image segmentation because *UMA* is very similar to *SS*.

For each image in the sizes subset, including its contrast and noise level varieties, 20 experiments were performed, the values to be used are the best obtained in the previous section for the circumference of a 64 radio (170 for the first contrast, 70 for the second and 4 for the third, depending on the noise levels), in the different contrasts of gray level.

In the first experiment performed, 4 algorithms are compared to the mentioned images and with a noise level of  $P = 1$  percent and  $\sigma = 80$ . In the second case the algorithms were evaluated with a noise level of  $P = 7.5$  percent and  $\sigma = 20$ . In the third case the algorithms were evaluated with a contamination level of  $P = 10$  percent and  $\sigma = 80$ . In the fourth case the algorithms were evaluated with a noise level of  $P = 25$  percent and  $\sigma = 40$ . In this paper the obtained results are shown for the two extreme cases.



**Fig. 3.** First comparison: Noise level  $P = 1$  percent and  $\sigma = 80$ . RG is Region Growing; T is Thresholding; NN is Neural Network; IS is the proposed algorithm.

### 4.3 Application of Evaluation and Comparison Results

In this section the previous results were applied to real images obtained from the database for image segmentation of Berkeley University, three images were chosen, the first two had a contrast between objects and background similar to the second contrast used in the sections of algorithm evaluation and comparison, for this reason the value to be used in this case was 70. The third image has

a difference between gray levels and background similar to the first contrast in previous experiments, for this reason a 170 value was used. In this paper the obtained results are shown for the two extreme cases.

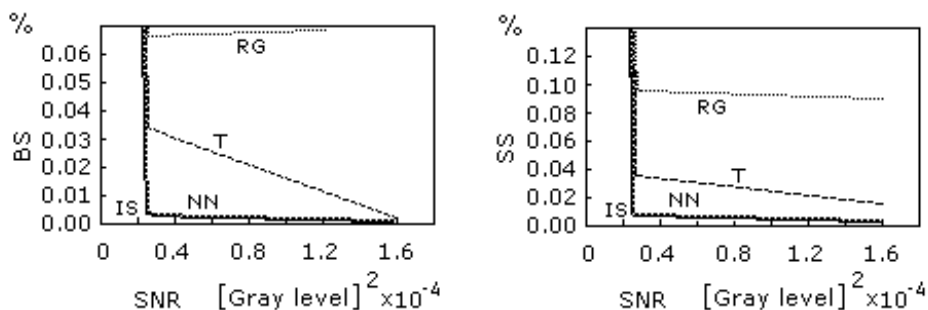
## 5 Conclusions and Future Projects

It was concluded that it is possible to develop an image segmentation method based on cluster analysis, and to find an evaluation and comparison technique for segmentation methods to validate it, which was verified with sets of objective indicators to evaluate and compare image segmentation algorithms, testing the treated images under different conditions.

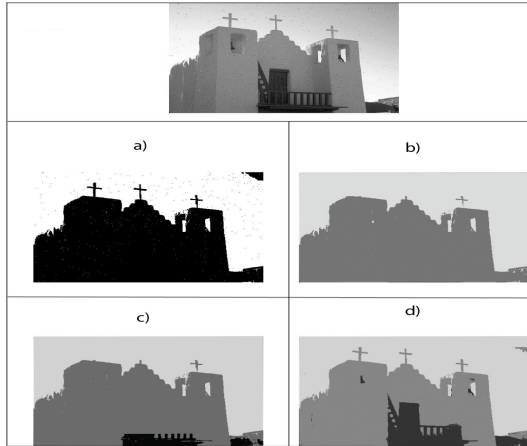
Also, based on the indicators that allow the algorithm evaluation, we were able to conclude that it is sensitive to the difference of gray levels within de image, since its behavior changes entirely depending on this feature. Another important conclusion in this aspect is that the algorithm acts better with higher differences among gray tonalities.

As the difference between gray tones increases, the value of  $T_{\text{means}}$  parameter also has to increase, and in case of very little contrasts the  $T_{\text{means}}$  value also has to behave accordingly, since that the algorithm provides more details during image segmentation.

The existence of different results depending on the shape was noted, this happens only in the highest contamination levels, for this reason it must be concluded that the algorithm is more sensitive to noise than to difference in shape within the images, which was also reflected in the value of performance indicator, since the performance indicator was higher as the noise in the image increased. In the case of the algorithm comparison, we can conclude that the main differences in algorithm behavior were due to the contrast variables among objects and background gray levels, and image noise levels. Based on this we can state that there is a great deficiency of thresholding when faced with low contrasts. In the case of the other three algorithms, they have similar behaviors

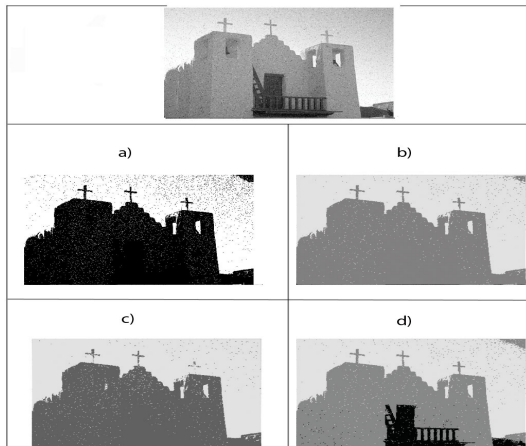


**Fig. 4.** Fourth comparison: Noise level of  $P = 25$  percent and  $\sigma = 40$ . RG is Region Growing; T is Thresholding; NN is Neural Network; IS is the proposed algorithm.



**Fig. 5.** Case 1: 1 percent contaminated image, standard deviation= 80. a) Thresholding; b) Neural Network; c) Region Growing; d) proposed algorithm.

to lower noise levels, having the best behavior the neuronal networks algorithm, followed by ISACA and finally Region Growing. As the contamination within the image increases independently of the gray levels contrast, an improvement is noted in the behavior of ISACA algorithm that exceeds the performance of the other three algorithms. Based on the comparison performed with real images obtained from the database for image segmentation of Berkeley University, we can conclude that the main differences that a more subjective evaluation (relying on the human eye) can see, so as to complement the proposed comparison, are



**Fig. 6.** Case 2: 25 percent contaminated image, standard deviation= 40. a) Thresholding; b) Neural Network; c) Region Growing; d) proposed algorithm.

that the ISACA algorithm, besides of having a better performance for higher noise levels, it has a better ability to perform a more detailed segmentation of the image. This is good depending on the problem, since in the segmentation image field problems are so wide-ranging that a definition of "good" must be given in terms of the approached problem. Finally we can conclude, based on the obtained results and the experience earned in this paper, that it really is possible to generate methods to evaluate and compare segmentation algorithms, but its performance will always be linked to the problem to be solved, this is why the general model addressed should be adapted to the problem.

## References

1. Allende, H. and Galbiati, J. (2004). A non-parametric to filter for digital image restoration, using cluster analysis. *Pattern Recognition Letters*, Vol 25, (June 2004), pp. 841-847.
2. Cortes, C. and Hertz, A. (1989). Network System for Image Segmentation *Intelligents Robotics*, 121-125.
3. Gonzalez, R. and Woods, R. (1992). *Digital Image Processing*. Adison-Wesley.
4. Haralick, R. and Sapiro, L. (1985). Image Segmentation Techniques. *Computer Vision, Graphics and Image Processing*, 29:100-132.
5. Ormoneit, D. and Tresp, V. (1998). Averaging, Maximum Penalized Likelihood and Bayesian Estimation for Improving Gaussian Mixture Probability Density Estimates *IEEE Transactions on Neural Networks*, 9:639-650.
6. Pal, N. and Pal, S. (1993). A Review on Image Segmentation Techniques. *Pattern Recognition*, 26:1277-1294. Pearson, D. (1991). *Visual Perception By Computer*. MacGraw-Hill.
7. Yang, L., Albregtsen, F., Lonnestad, T. and Grottum, P. (1999). A Supervised approach to the Evaluation of Image Segmentation Methods Department of Informatics, University of Oslo
8. Zhang, Y. (1996). A survey on Evaluation Methods for Image Segmentation. *Journal of Automated Reasoning*, 29:1335-1346.

# Active Shape Models and Evolution Strategies to Automatic Face Morphing

Vittorio Zanella, Héctor Vargas, and Lorna V. Rosas

Universidad Popular Autónoma del Estado de Puebla  
21 sur 1103 Col. Santiago 72160 Puebla Pue. México  
{vittorio.zanella, hectorsimon.vargas,  
lornaveronica.rosas}@upaep.mx

**Abstract.** Image metamorphosis, commonly known as morphing, is a powerful tool for visual effects that consists of the fluid transformation of one digital image into another. There are many techniques for image metamorphosis, but in all of them there is a need for a person to supply the correspondence between the features in the source image and target image. In this paper we use the Active Shape Models and Evolution Strategies to perform the metamorphosis of face images in frontal view automatically.

## 1 Introduction

Image metamorphosis is a powerful tool for visual effects that consists of the fluid transformation of one digital image into another. This process, commonly known as *morphing* [1], has received much attention in recent years. This technique is used for visual effects in films and television [2], [3], and it is also used for recognition of faces and objects [4].

Image metamorphosis is performed by coupling image warping with color interpolation. Image warping applies 2D geometric transformations to images to retain geometric alignment between their features, while color interpolation blends their colors.

The quality of a morphing sequence depends on the solution of three problems: feature specification, warp generation and transition control. Feature specification is performed by a person who chooses the correspondence between pairs of feature primitives. In actual morphing algorithms, meshes [5], [6], line segments [7], [8], [9], or points [10], [11], [12] are used to determine feature positions in the images. Each primitive specifies an image feature, or landmark. Feature correspondence is then used to compute mapping functions that define the spatial relationship between all points in both images. These mapping functions are known as warp functions and are used to interpolate the positions of the features across the morph sequence. Once both images have been warped into alignment for intermediate feature positions, ordinary color interpolation (cross-dissolve) is performed to generate image morphing. Transition control determines the rate of warping and color blending across the morph sequence.

Feature specification is the most tedious aspect of morphing, since it requires a person to determine the landmarks in the images. Many methods have been developed

to extract facial features. Most of them are based on neural nets [13], geometrical features of images [14], and template matching methods [15]. Unfortunately, most methods require significant human participation. A way to determine the landmarks automatically, without the participation of a human, would be desirable. In this work, we use the Active Shape Models and Evolution Strategies to find the facial features and the spatial relationship between all points in both images, without the intervention of a human expert.

## 2 Active Shape Models

The Active Shape Models (ASM) [16] are statistical models which iteratively move toward structures in images similar to those on which they were trained. The aim is to build a model that describes shapes and typical variations of an object. In this case the object is a face. To make the model able of capturing typical variations we use different face images appearing in different ways reflecting its possible variations. This set of images is named the training set. In this work, the trained set comprises 37 different frontal human image faces, all without glasses and with a neutral expression [17]. To collect information about the shape variations necessary to build the model, we represent each shape with a set of 58 landmarks points. Each image in the training set is labeled with the set of points; each labeled point represents a particular part of the face or its boundary. Each point will thus have a certain distribution in the image space.

## 3 Point Distribution Model

The Point Distribution Model is generated from the examples of faces shapes, where each face is represented by a set of labeled points. A given point corresponds to a particular location on each face to be modeled [18]. The examples faces are all aligned into a standard co-ordinate frame, and a principal component analysis is applied to the coordinates of the points. This produces the mean position for each of the points and description of the main ways in which the points tend to move together. The model can be used to generate new face shape using the equation

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \quad (1)$$

where  $\mathbf{x} = (x_0, y_0, \dots, x_{n-1}, y_{n-1})^T$ ,  $(x_k, y_k)$  is the  $k^{th}$  model point.

$\bar{\mathbf{x}}$  represents the mean shape

$\mathbf{P}$  is a  $2n \times t$  matrix of  $t$  unit eigenvectors

$\mathbf{b} = (b_1, \dots, b_t)^T$  is a set of shape parameters  $b_i$

If the shape parameters  $b_i$  are chosen such that the square of the Mahalanobis distance  $D_m^2$  is limited, then the shape generated by (1) will be similar to those given in the training set

$$D_m^2 = \sum_{k=1}^t \left( \frac{b_k^2}{\lambda_k} \right) \leq D_{\max}^2 \quad (2)$$

$\lambda_k$  is the variance of parameter  $b_k$  in the original training set and  $D_{\max}^2 = 3.0$ . By choosing a set of shape parameters  $\mathbf{b}$  for a Point Distribution Model, we define the shape of a model object in an object centred coordinate frame. We can then create an instance,  $\mathbf{X}$ , of the model in the image frame by defining the position, orientation and scale:

$$\mathbf{X} = M(s, \theta)[\mathbf{x}] + \mathbf{X}_c \quad (3)$$

where  $\mathbf{X}_c = (X_c, Y_c, \dots, X_c, Y_c)^T$ ,  $M(s, \theta)[\ ]$  performs a rotation by  $\theta$  and scaling by  $s$ , and  $(X_c, Y_c)$  is the position of the centre of the model in the image frame.

## 4 Modeling Grey Level Appearance

We wish to use our models for locating facial features in new face images. For this purpose, not only shape, but also grey-level appearance is important. We account for this by examining the statistics of the grey levels in regions around each of the labelled model points [19]. Since a given point corresponds to a particular part of the object, the grey-level patterns about that point in images of different examples will often be similar. We need to associate an orientation with each point of our shape model in order to align the region correctly, in this case normal to the boundary. For every point  $i$  in each image  $j$ , we can extract a profile  $g_{ij}'$ , of length  $n_p$  pixels, centred at the point. We choose to sample the derivative of the grey levels along the profile in the image and normalise.

If the profile runs from  $p_{start}$  to  $p_{end}$  and is of length  $n_p$  pixels, the  $k^{th}$  element of the derivative profile is

$$g_{jk}' = I_j(\mathbf{y}_{k+1}) - I_j(\mathbf{y}_{k-1}) \quad (4)$$

where  $\mathbf{y}_k$  is the  $k^{th}$  point along the profile:

$$\mathbf{y}_k = \mathbf{p}_{start} + \frac{k-1}{n_p-1}(\mathbf{p}_{end} - \mathbf{p}_{start}) \quad (5)$$

and  $I_j(\mathbf{y}_k)$  is the grey level in image  $j$  at that point. We then normalise this profile,

We can then calculate an  $n_p \times n_p$  covariance matrix,  $\mathbf{S}_{g_i}$ , giving us a statistical description of the expected profiles about the point. Having generated a flexible model and a description of the grey levels about each model point we would like to find new examples of the modelled face in images.

## 5 Calculating a Suggested Movement for Each Model Point

Given an initial estimate of the positions of a set of model points which we are attempting to fit to a face image we need to estimate a set of adjustments which will move each point toward a better position. At a particular model point we extract a derivative profile  $\mathbf{g}$ , from the current image of some length  $l$  ( $l > n_p$ ), centered at the point and aligned normal to the boundary. We then run the profile model along this

sampled profile and find the point at which the model best matches. Given a sampled derivative profile the fit of the model at a point  $d$  pixels along it is calculated as follows [19];

$$f_{prof}(d) = (\mathbf{h}(d) - \bar{\mathbf{g}})^T \mathbf{S}_{\mathbf{g}}^{-1} (\mathbf{h}(d) - \bar{\mathbf{g}}) \quad (6)$$

where  $\mathbf{h}(d)$  is a sub-interval of  $\mathbf{g}$  of length  $n_p$  pixels centred at  $d$ , and normalised. This is the Mahalanobis distance of the sample from the mean grey model, the value of  $f_{prof}$  decreases as the fit improves. The point of best fit is thus the point at  $f_{prof}(d)$  is minimum [20].

## 6 Estimates the Initial Position and Size of Model Points

The estimation of the initial position is performed using evolution strategies that are algorithms based on Darwin's theory of natural evolution, which states that only the best individuals survive and reproduce. Our algorithm is a (1+1)-ES algorithm, i.e. the initial population is formed only by one individual (a face model) then mutation is the only operation utilized [21]. The form of the individual is  $i = ((x_1, y_1), (x_2, y_2), \dots, (x_{58}, y_{58}))$  that corresponds the 58 points in the model. The mutation operation corresponds to a one affine transformation with parameters  $s_x, s_y, t_x, t_y$ , where  $s_x$  and  $s_y$  are the scale parameters in  $x$  and  $y$  respectively, and  $t_x$  and  $t_y$  are the translation parameters in  $x$  and  $y$ . Rotation in this case is not used because we assume that the images are in frontal view and not rotated.

The mutation operation consists of modifying the translation and scale parameters. It adds normal random numbers with mean  $\mu$  and standard deviation  $\sigma$ ,  $N(\mu, \sigma)$ , in the following way:

$$t'_x = t_x + N(0,1) \quad t'_y = t_y + N(0,1) \quad (7)$$

$$s'_x = s_x * N(1,0.5) \quad s'_y = s_y * N(1,0.5) . \quad (8)$$

Scale and translation are performed using the following matrix:

$$S = \begin{pmatrix} s'_x & 0 & 0 \\ 0 & s'_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t'_x & t'_y & 1 \end{pmatrix} .$$

We perform the following operations for each point in the model

$$(x', y', 1) = (x, y, 1) \cdot S \text{ to scale, and } (x', y', 1) = (x, y, 1) \cdot T \text{ to translate.} \quad (9)$$

For the fitness function we need to find the binary image,  $\phi(I)$ , corresponding to the source image. We use the fact that the image have uniform illumination; in this case most of the points in the image depict skin and then we can find the regions in



the face that are darker than skin, for example the eyes or mouth. Once we have the binary image, we compute the following function:

$$Fitness = \max(A_{eyes} + A_{mouth}) \tag{10}$$

Where

$$A_{eyes} = \iint_{R_{El}} (\phi(I)) dI + \iint_{R_{Er}} (\phi(I)) dI \quad \text{and} \quad A_{mouth} = \iint_{R_M} (\phi(I)) dI \tag{11}$$

$R_{El}$ ,  $R_{Er}$  and  $R_M$ , correspond to the left eye, right eye and mouth regions respectively.

## 7 Warp Generation

Once the model has been adjusted to the images, the next step is to perform image deformation, or warping, by mapping each feature in the source image to its correspondent feature in the target image. When we use point-based feature specification, we must deal with the problem of scattered data interpolation.

The problem of scattered data interpolation is to find a real valued multivariate function interpolating a finite set of irregularly located data points. For bivariate functions, this can be formulated as [22]:

*Input:*  $n$  data points  $(x_i, y_i)$ ,  $x_i \in \mathfrak{X}^2$ ,  $y_i \in \mathfrak{Y}$ ,  $i=1, \dots, n$ .

*Output:* A continuous function  $f: \mathfrak{X}^2 \rightarrow \mathfrak{Y}$  interpolating the given data points, i.e.  $f(x_i) = y_i$ ,  $i = 1, \dots, n$ .

We use the inverse distance weighted interpolation method described in [22].

## 8 Transition Control

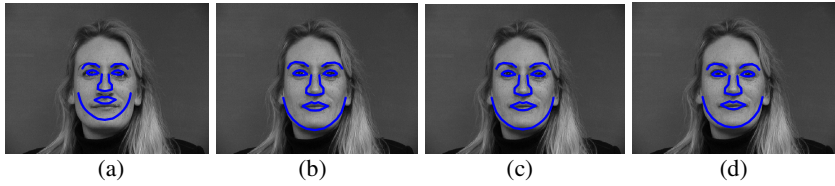
For obtain the transition between the source image and the target image we use linear interpolation of their attributes. If  $I_S$  and  $I_T$  are the source and target images we generate the sequence of images  $I_\lambda$ ,  $\lambda \in [0,1]$ , such that

$$I_\lambda = (1 - \lambda) \cdot I_S + \lambda \cdot I_T \tag{12}$$

This method is called cross-dissolve.

## 9 Results

The results in the phase of the feature localization are shown in Fig. 1. In Fig. 2 and Fig. 3 is showed two examples of the morphing between two faces, in this figures the first and third rows show the warping of source and target images respectively and the second row show the morphing process using cross-dissolve.



**Fig. 1.** The individual adjusted to the image (a) Initial model , (b)After 3 iterations, (c) After 5 iterations and (d) After 10 iterations



**Fig. 2.** Morphing example 1



**Fig. 3.** Morphing example 2

## 10 Conclusions

We developed a method to perform the morphing between two face images automatically. The facial features localization was performed using Active Shape Models and Evolution Strategies. The warp generation was performed using the inverse distance weighted interpolation method for the problem of scattered data interpolation and used cross-dissolve for transition control. The method work automatically once the training set was labeled with the set of points, and this work is made for a person. We work only with face images in frontal view.

## References

1. G. Wolberg, Image Morphing: a Survey, *The Visual Computer* Vol. 14, 360-372, 1998
2. P. Litwinowicz, & L. Williams, Animating Images with Drawings, *Proceedings of the SIGGRAPH Annual Conference on Computer Graphics*, 409-412, 1994
3. G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, Los Alamitos CA., 1990
4. Bichsel, Automatic Interpolation and Recognition of Face Images by Morphing, *The 2<sup>nd</sup> International Conference on Automatic Face and Gesture Recognition*. IEEE Computer Society Press, Los Alamitos, CA, 128-135, October 1996
5. W. Aaron, et al., Multiresolution Mesh Morphing, *Proceedings of the SIGGRAPH Annual Conference on Computer Graphics*, 343-350, August 1999.
6. G. Wolberg, Recent Advances in Image Morphing, *Computer Graphics International*, Pohang Korea, June 1996
7. T. Beier, & N. Shawn, Feature-Based Image Metamorphosis, *Proceedings of the SIGGRAPH Annual Conference on Computer Graphics*, Vol. 26, No. 2, 35-42, July 1992.
8. S. Lee, K. Chwa, & S. Shin, Image Metamorphosis Using Snakes and Free-Form Deformations, In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 439-448. ACM SIGGRAPH, Addison Wesley, August 1995.
9. S. Lee et al., Image Metamorphosis with Scattered Feature Constraints, *IEEE Transactions on Visualization and Computer Graphics*, 2:337--354, 1996.
10. Nur, et al., Image Warping by Radial Basis Functions: Applications to Facial Expressions, *CVGIP: Graph Models Image Processing*, Vol.56, No. 2, 161-172, 1994
11. S. Lee, et al., Image Morphing Using Deformable Surfaces, *Proceedings of the Computer Animation Conference*, IEEE Computer Society, 31-39, May 1994
12. S. Lee, et al., Image Morphing Using Deformation Techniques, *J. Visualization Comp. Anim.* No.7, 3-231, 1996
13. N. Intrator, D. Reisfeld, & Y. Yeshurun, Extraction of Facial Features for Recognition using Neural Networks, *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, Zurich, 260-265, 1995.
14. Craw, D. Tock and A. Bennett, Finding Face Features, *Proceedings of the European Conference on Computer Vision*, ECCV-92, Ed. G. Sandini, 92-96, Springer-Verlag 1992
15. M. J. T. Reinders. Model Adaptation for Image Coding. PhD thesis, Delft University of Technology, Delft, The Netherlands, Dec. 1995.
16. T. Cootes, C. Taylor, D. Cooper, J. Graham, Active Shape Models- Their Training and Application, *Computer Vision and Image Understanding*, Vol. 61, No. 1, January, 1995, pp. 38-59.

17. M.B. Stegmann, Analysis and Segmentation of Face Images using Point Annotations and Linear Subspace Techniques, Informatics and Mathematical Modelling, Technical University of Denmark, IMM-REP-2002-xx, 2002.
18. T. Cootes, A. Hill, C. Taylor and J. Haslam, Use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6): 1994, 355-366.
19. T. Cootes, C. Taylor, A. Lanitis, D. Cooper, J. Graham, Building and Using Flexible Models Incorporating Grey-Level Information, Proc. 4<sup>th</sup>. ICCV, IEEE Computer Society Press 1993, pp. 242-246.
20. T. Cootes, C. Taylor, A. Lanitis, D. Active Shape Models: Evaluation of a Multiple-Resolution method for improving Image Search, *Proc. British Machine Vision Conference*, Vol. 1, 1994, pp. 327-336
21. Zbigniew Michalewicz, Genetic Algorithms+Data Structures=Evolution Programs, Springer-Verlag, ISBN: 3-540-60676-9, 1999.
22. D. Ruprecht and H. Muller, Image warping with scattered data interpolation. *IEEE Computer Graphics and Applications*, 15( 2), 37-43. 1995.

# Recognition of Shipping Container Identifiers Using ART2-Based Quantization and a Refined RBF Network

Kwang-Baek Kim<sup>1</sup>, Minhwan Kim<sup>2</sup>, and Young Woon Woo<sup>3</sup>

<sup>1</sup> Dept. of Computer Engineering, Silla University, Busan, Korea  
gbkim@silla.ac.kr

<sup>2</sup> Dept. of Computer Engineering, Pusan National University, Busan, Korea  
mhkim@pusan.ac.kr

<sup>3</sup> Dept. of Multimedia Engineering, Dong-Eui University, Busan, Korea  
ywoo@deu.ac.kr

**Abstract.** Generally, it is difficult to find constant patterns on identifiers in a container image, since the identifiers are not normalized in color, size, and position, etc. and their shapes are damaged by external environmental factors. This paper distinguishes identifier areas from background noises and removes noises by using an ART2-based quantization method and general morphological information on the identifiers such as color, size, ratio of height to width, and a distance from other identifiers. Individual identifier is extracted by applying the 8-directional contour tracking method to each identifier area. This paper proposes a refined ART2-based RBF network and applies it to the recognition of identifiers. Through experiments with 300 container images, the proposed algorithm showed more improved accuracy of recognizing container identifiers than the others proposed previously, in spite of using shorter training time.

## 1 Introduction

Identifiers of a shipping container are given in accordance with the terms of ISO standard, which consist of 4 code groups such as shipping company codes, container serial codes, check digit codes, and container type codes. Only the first 11 identifier characters are prescribed in the ISO standard and shipping containers can be discriminated by automatically recognizing them [1]. However the ISO standard prescribes only the code type on container identifiers, so other features such as the foreground and background colors, the font type, the size and the position of container identifiers, etc., vary from one container to another. Sometimes shapes of identifiers can be also impaired by the environmental factors during the transportation by sea, since the identifiers are just printed on the surface of a container. Furthermore, the damage to a container surface or image noises may lead to distortion of shapes of identifier characters in a container image. Such variations and distortions make it quite difficult to extract and recognize the identifiers using simple morphological information like shape, size, and position [2].

In the preprocessing of a container image for the extraction of container identifiers, it is necessarily required to distinguish whether extraction results are contours of identifiers or background noises. In this paper, at first, color information of a container image is clustered by using the ART2 algorithm [3] and is quantized based on the predefined bin vectors, then the 8-directional contour tracking method [4] is applied to the quantized image. By using feature information of container identifiers, background noises are removed and identifiers are extracted from the areas labeled by the contour tracking method.

This paper proposed a refined ART2-based RBF network and applied it to the identifier recognition. In the proposed ART2-based RBF network, the refined ART2 algorithm is applied to the middle layer of RBF network, which dynamically adjusts the vigilance parameter by using the fuzzy logic connection operator, and the learning ratio is dynamically adjusted by applying the delta-bar-delta method to the learning between the middle and the output layers of RBF network.

## 2 Extraction of Identifier Areas and Individual Identifiers

In this paper, the extraction process of container identifiers quantizes a container image, extracts identifier areas from the quantized image, binarizes the extracted areas and last, extracts individual identifiers from the binarized areas.

### 2.1 ART2-Based Quantization of a Container Image

Color information of a container image is able to be classified by using an image quantization method. Generally, it may occur in the image quantization the problem that two pixels having similar color information at the neighborhood of quantization boundary are classified to different bins, incurring the loss of effective information. This paper, to refine the problem, clusters color information of a container image by using ART2 algorithm and quantizes the container image based on the similarity between the predefined  $n$  color bins and the centers of clusters.

For the quantization of a container image, at first, this paper classifies similar color vectors to a cluster by apply ART2 algorithm to a container image. The input patterns of ART2 algorithm are three-dimensional vectors indicating color coordinates in RGB space, and the center vector of a cluster is the mean vector of color vectors included in the cluster. Fig. 1(b) shows the result of ART2-based clustering of an original container image, Fig. 1(a).

For the optimal quantization of a container image in RGB color space, a standard bin group with  $n$  elements defined as (R, G, B) vectors is given in advance. After calculating the similarity between the center vector of a cluster and each bin vector like Eq. (1), the bin vector with the highest similarity is selected as a quantization code, and color vectors included in the cluster are quantized by using the selected bin vector.

$$S(x, y) = \alpha \cdot \left( 1 - \sum_{i=1}^n (x_i - y_i)^2 \right) + \beta \cdot \left( 1 - \frac{|x \cdot y|}{\sqrt{|x||y|}} \right) \quad (1)$$

where  $x$  and  $y$  are three-dimensional vectors in RGB space and  $\alpha$  and  $\beta$  are parameters adjusting the ratio between the Euclidean distance and color values. Fig. 1(c) shows the quantized image of Fig. 1(a).



**Fig. 1.** An original container image and the processed results

## 2.2 Extraction of Identifier Areas

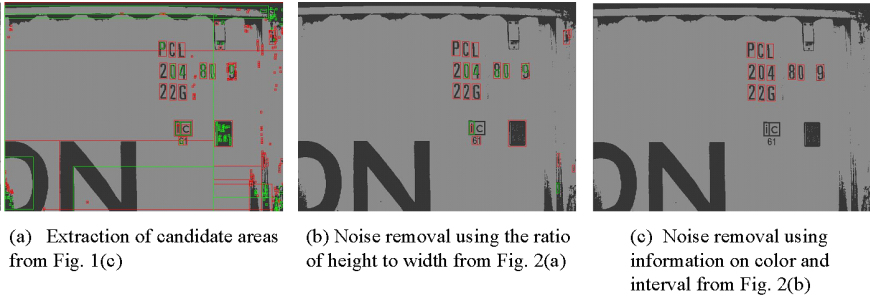
An identifier area means the rectangle area including only container identifiers in a container image. This paper, at first, extracts candidate areas for identifier areas by applying the 8-directional contour tracking method to the quantized image, and next, selects target areas among candidate ones by using two types of noise-removal method which remove areas corresponding to background noises.

Although the size of a container identifier is not constantly prescribed, the ratio of height to width is kept to be constant. So, the first type of noise-removal method removes noise areas by using the ratio of height to width. Fig. 2(a) shows candidates for identifier areas extracted from Fig. 1(c) and Fig. 2(b) shows the result image obtained by applying the first-type noise-removal method to Fig. 2(a).

Container identifiers have similar colors and arranges in the vertical or horizontal direction on the container surface, keeping a constant interval from other ones. In the second type of noise-removal method, after measuring the distance between candidate areas, areas being apart over the given interval from other ones may be removed as noises, and identifier areas are selected from remaining candidates by using color information. Fig. 2(c) shows the noised-removed quantized image obtained to applying the second-type noise-removal method to Fig. 2(b) in succession.

## 2.3 Binarization of Identifier Areas and Extraction of Individual Identifiers

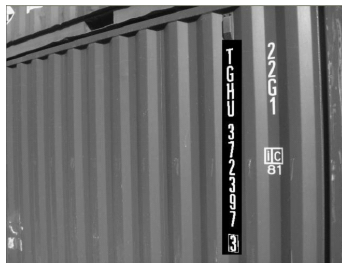
After converting identifier areas to grayscale areas, the iterative binarization method is applied to converted areas for image binarization. In a container image,



**Fig. 2.** Processes for extraction of candidate areas and noise removal

bends exit in the horizontal direction, and the distortion of color information occurs in areas shaded by the bends. Since container identifiers arrange in the vertical or horizontal direction, identifier areas including horizontally-arranged identifiers may contain noises caused by shadows of bends. So, considering the arrangement direction of identifiers, the binarization method has to be applied by the different schemes.

This paper determines the arrangement direction of identifiers by measuring the ratio of the number of horizontally-arranged areas to vertically-arranged ones. Vertically-arranged identifier areas are binarized by applying the iterative binarization method once in the vertical direction. For horizontally-arranged areas, to remove noises caused by bends, the iterative method is applied twice in the vertical and the horizontal directions separately and two outputs are combined by using AND image operation to one binarized area. Fig. 3 shows the binarization result of vertically-arranged identifier areas. Fig. 4(a) shows the binarization result obtained by applying the iterative method in the vertical direction to horizontally-arranged areas, indicating that the simple binarization scheme for horizontally-arranged areas may be failed due to noises by bends. On the other hand, Fig. 4(b) shows that the proposed scheme for horizontally-arranged areas is successful in the binarization.



**Fig. 3.** Binarization result of vertically-arranged identifier areas



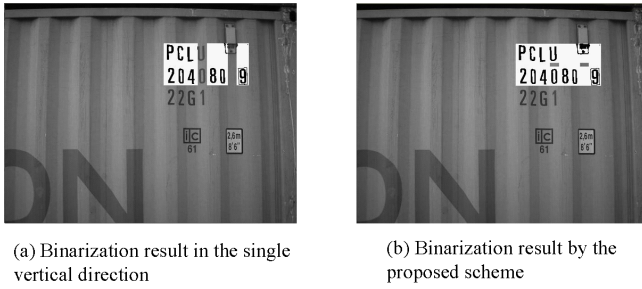


Fig. 4. Binarization result of horizontally-arranged identifier

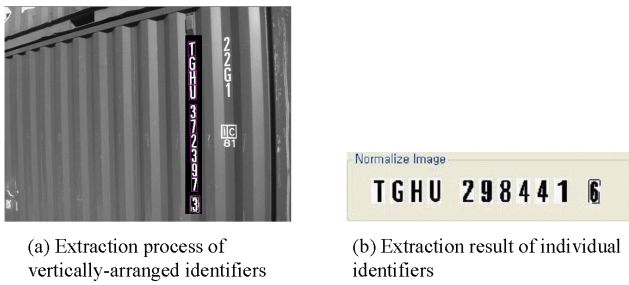


Fig. 5. Extraction of individual identifiers in vertically-arranged areas

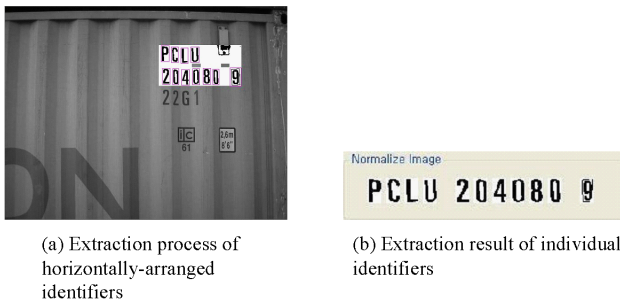


Fig. 6. Extraction of individual identifiers in horizontally-arranged areas

Individual identifiers are extracted by applying 8-directional contour tracking method to identifier areas, generating pixel areas corresponding to 11 prescribed identifier codes. Fig. 5 and Fig. 6 show the extraction results of individual identifiers in vertically-arranged areas and horizontally-arranged ones, respectively.

### 3 Identifier Recognition Using a Refined ART2-Based RBF Network

This paper proposed a refined ART2-based RBF network for the recognition of container identifiers. In a conventional ART2 algorithm, vigilance parameters are heuristically fixed, causing the problem that similar patterns are classified to different clusters or different patterns are classified to the same cluster [5].

This paper, first, proposed the refined ART2 algorithm having the learning structure that dynamically adjusts vigilance parameters by using fuzzy logic intersection operator, selects a node with minimum output value as a winner node and transmits the winner node to the output layer. And the refined ART2 algorithm was applied to the learning between the input and the middle layers in RBF network. Also, in the proposed RBF network, the generalized delta learning method was applied to the learning between the middle and the output layers and delta-bar-delta algorithm was used to improve the performance of learning [6]. The learning algorithm of the refined ART2-based RBF network is summarized as follows:

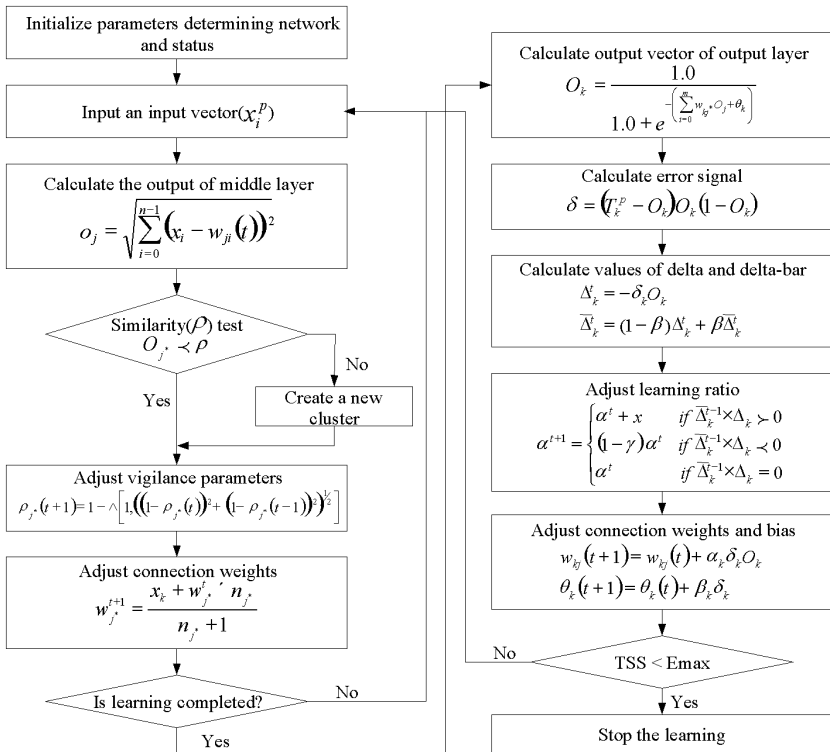


Fig. 7. Refined ART2-based RBF network

1. The competitive learning adjusting dynamically the learning rate is performed between the input and the middle layers by applying the refined ART2 algorithm.
2. Nodes of the middle layer mean individual classes. Therefore, while the proposed RBF network has a fully-connected structure on the whole, it takes the winner node method that compares target vectors and output vectors and back-propagates only the connection weight to the representative class.
3. The proposed RBF network performs the supervised learning by applying the generalized delta learning to the learning structure between the middle and the output layers.
4. The proposed RBF network improves the performance of learning by applying delta-bar-delta algorithm to the generalized Delta learning for the dynamical adjustment of a learning rate. When defining the case that the difference between the target vector and the output vector is less than 0.1 as accuracy and the opposite case as inaccuracy, Delta-bar-Delta algorithm is applied restrictively in the case that the number of accuracies is greater than or equal to inaccuracies with respect to total patterns. This prevents no progress or an oscillation of learning keeping almost constant level of error by early premature situation incurred by competition in the learning process.

The detailed description of the refined ART2-based RBF network is like Fig. 7.

### 4 Performance Evaluation

The proposed algorithm was implemented by using Microsoft Visual C++ 6.0 on the IBM-compatible Pentium-IV PC for performance evaluation. 300 container images with size of 640x480 were used in the experiments for extraction and recognition of container identifiers. The implemented output screen of identifier extraction and recognition is like Fig. 8.

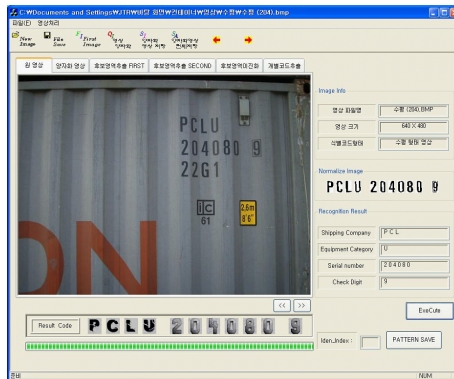


Fig. 8. Experiment screen of extraction and recognition of container identifiers

In the extraction of identifier areas, the previously proposed method in Ref. [7] fails to extract target areas due to noises caused by an external light and the rugged surface shape of containers. On the other hand, the proposed extraction method detects and removes noises by using ART2-based image quantization method and feature information of container identifiers, improving the success rate of extraction compared with the previously proposed. The comparison of the success rate of identifier area extraction between the proposed in this paper and the previously proposed in Ref. [7] is like Table 1.

**Table 1.** Comparison of the success rate of identifier area extraction

	Previously proposed method in Ref. [7]	The proposed method in this paper
Success rate	209/300 (69.6% )	282/300 (94% )

For the experiment of identifier recognition, by applying 8-directional contour tracking method to 282 identifier areas extracted by the proposed extraction algorithm, a total of 3102 identifier codes were extracted. The extracted codes consisted of 1128 shipping company codes, 1692 container serial codes and 282 check digit codes. This paper performed the learning in the proposed RBF network using 200 container serial codes and 480 shipping company codes and 100 check digit codes as learning data, and the experiment for identifier recognition was performed using all extracted identifier codes. Table 2 shows the performance of learning and recognition of the proposed RBF network.

Fig. 9 shows the change process of TSS according to the number of Epochs with respect to each type of container identifiers in the refined ART2-based RBF network. As shown in Fig. 9, the proposed RBF network is fast in the initial convergence and performs the stable learning. In the experiment for performance evaluation, parameter setup for the proposed RBF network was like Table 3. In Table 3,  $\rho$ ,  $\alpha$  and  $\mu$  mean the vigilance parameter, the learning rate and the momentum coefficient, respectively, and  $\kappa$ ,  $\gamma$  and  $\beta$  are delta-bar-delta constants.

**Table 2.** Performance of learning and recognition in the proposed RBF network

	Refined ART2-based RBF network	
	# of Epoch	# of success of recognition
Shipping Company Codes (1128)	1170	1126
Container Serial Codes (1692)	669	1681
Check Digit Codes (282)	319	282

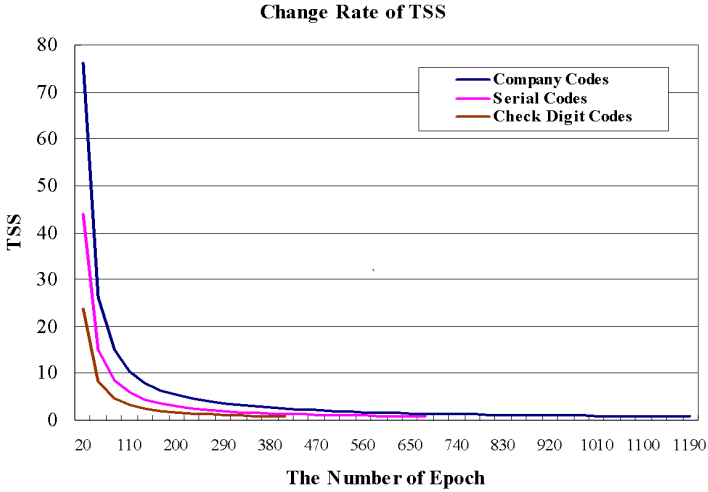


Fig. 9. Change process of TSS according to the number of Epochs

Table 3. Parameter setup for the refined ART2-based RBF network

Parameters	$\rho$	$\alpha$	$\mu$	$\kappa$	$\gamma$	$\beta$
Refined ART2-based RBF network	0.05	0.8	0.6	0.03	0.2	0.7

The refined ART2-based RBF network failed to recognize container identifiers in the cases that individual identifiers are largely damaged in an original container image or the loss of information on identifiers occurs in the binarization phase of identifier areas.

## 5 Conclusions

This paper, at first, distinguished identifier areas from background noises and removed noises by using ART2-based quantization method and morphological information on container identifiers such as color, size, ratio of height to width and an interval between identifiers, etc. And, using 8-directional contour tracking method, individual identifiers were extracted from identifier areas. This paper proposed a refined ART2-based RBF network and applied to the recognition of individual identifiers.

Experiments using 300 container images showed that 282 areas of identifiers and 3102 individual identifiers were extracted successfully. The proposed RBF network recognized 3089 identifiers. Failures of recognition were caused by the damage of shapes of individual identifiers in original images and the information loss on identifiers shaded by bends in the binarization process.

A Future work is the development of fuzzy association algorithm that may recover damaged identifiers to improve the performance of extraction and recognition of individual identifiers.

## References

1. Freight Containers-Coding, Identification and marking [ISO 6346 1995(E)]
2. Kim, K. B.: Recognition of Identifiers from Shipping Container Images using Fuzzy Binarization and Neural Network with Enhanced Learning Algorithm. *Applied Computational Intelligence*. World Scientific. (2004) 215–221
3. Donna, L. H., Maurice, E. C.: *Neural Networks and Artificial Intelligence for Biomedical Engineering*. IEEE Press. (2000)
4. Jain, R., Kasturi, R., Schunck, B. G.: *Machine Vision*. McGraw-Hill, Inc. (1995)
5. Kim, K. B., Kim, C. K.: Performance Improvement of RBF Network using ART2 Algorithm and Fuzzy Logic System. *Lecture Notes in Artificial Intelligence, LNAI 3339*, Springer (2004) 853–860
6. Kim, K. B., Lee, L. U., Sim, K. B.: Performance Improvement of Fuzzy RBF Networks. *Lecture Notes in Computer Science, LNCS 3610* (2005), 237–244
7. Nam, M. Y., Lim, E. K., Heo, N. S., Kim, K. B.: A Study on Character Recognition of Container Image using Brightness Variation and Canny Edge. *Proceedings of Korea Multimedia Society*, 4-1 (2001) 111–115

# A Local-Information-Based Blind Image Restoration Algorithm Using a MLP

Hui Wang<sup>1</sup>, Nian Cai<sup>2</sup>, Ming Li<sup>1</sup>, and Jie Yang<sup>1</sup>

<sup>1</sup>Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University,  
Shanghai 200240, China

{hui, mingli, jieyang}@sjtu.edu.cn

<sup>2</sup>Information Engineering School, Guangdong University of Technology,  
Guangzhou 510006, China

cainian918@yahoo.com.cn

**Abstract.** Based on a multilayer perceptron (MLP), a blind image restoration method is presented. The algorithm considers both local region information and edge information of an image. To reduce the dimension of the network's input, a sliding window approach is employed to extract the features of the blurred image, which makes use of local region information. For the purpose of accelerating training and improving the restoration performance, the edge part and the smooth part in an image are separated and then used as training sets, respectively. A mapping model between the blurred image and the clear one is established through training the MLP with LM algorithm and then it is utilized to restore the blurred image. The simulation results demonstrate the proposed method feasible for image restoration.

## 1 Introduction

Image restoration is a process of removal or minimization of degradation from an image distorted by blurring and additive noise [1]. Image degradation not only decays the quality of the image, but also makes the image lose some important information, which will make further image processing difficult or even impossible. Therefore, the restoration of degraded image is an important and widely studied problem in computer vision and image processing. Various image restoration methods have been proposed [2] [3] [4] [5]. Blind deconvolution and Wiener filtering are two classical approaches used for image restoration. However, they might cause ill-posed problem and the traditional methods are deficient in solving the problem of function approximation.

Artificial neural network has its unique advantages in this aspect, for it is a type of large-scale nonlinear dynamical system, characteristic of high-speed parallelism calculation, great robustness, strong capacity of self-adaptive, self-organization and self-learning [6]. Some promising results on image restoration using neural networks have been reported in literatures [7] [8] [9]. Up to date, the most widely used neural network models for image restoration tasks are the Hopfield and the feed-forward neural networks, for example, the Multilayer Perceptron (MLP). The Hopfield network is used to formulate image restoration as a nonlinear optimization problem,

but it requires that the Point Spread Function (PSF) is known. However, in practical application, this requirement is impossible to satisfy while MLP intends to approximate a nonlinear filter through examples learning and it does not need this constraint.

In this paper, we propose a new method for blind image restoration using a three-layer MLP, combining the local information, which are region statistic information and edge information. The aim of the algorithm is to restore the original image from the blurred one without a prior knowledge about the PSF by exploiting the generalization capabilities of the MLP. A sliding-window technique is applied to obtain the features of the blurred image for dimension reduction. For smooth region and edge region, the training samples are selected separately and the MLP is employed to realize the obscure functional mapping from the degraded image space to the original image space.

## 2 Image Degradation Model

The discrete convolution degradation of 2-D image is modeled as:

$$g(i, j) = f(i, j) * h(i, j) + n(i, j) \quad (1)$$

where  $*$  denotes the convolution operation,  $g(i, j)$ ,  $f(i, j)$  and  $n(i, j)$  are the 2-D degraded observed image, original image and noise respectively.  $h(i, j)$  is the point spread function (PSF) operator. The multiplication model driven from the components of Eq.(1) are described as:

$$g = Hf + n \quad (2)$$

where  $g$ ,  $f$  and  $n$  are lexicographically ordered vectors of  $g(i, j)$ ,  $f(i, j)$  and  $n(i, j)$  respectively, and  $H$  is block circulant matrix of special distribution of the elements of  $h(i, j)$ .

## 3 Image Restoration Using a MLP

It has been shown that the MLP with a single hidden layer and a sigmoid function as excitation function can approximate any continuous nonlinear mapping on a compact set. [10]. Therefore, a standard three-layer MLP is used in our experiments. The structure of the network is determined as follows. The number of the input nodes is nine, which equals to the dimension of the input pattern, and the number of the output nodes is one, which represents the corresponding pixel in the target or the restored image. The initial hidden node number could be calculated via [11]:

$$n1 = \sqrt{n + m} + a \quad (3)$$

where  $n1$  is the number of hidden nodes;  $n$  is the number of input nodes;  $m$  is the number of output nodes; and  $a$  is a constant between 1 and 10. In order to get a better performance, we will determine a suitable number of the hidden-layer nodes  $n2$ , based on  $n1$ , by checking the square error when it is varied from a small value (say,



three) to a reasonably big value (say, thirty). For each chosen number, the neural network is trained, and the averaged error on the validation sets is estimated. After the above procedure is repeated for every number, the number of hidden nodes that produces the smallest averaged error on the validation sets is used. The model precision was tested with the varying  $n_2$  until the errors became stable. Here 20 hidden nodes were established for the highest convergence rate to the desired accuracy (data not shown). Thus the MLP with topological structure 9:20:1 was established for image restoration. The hidden nodes use the sigmoid transfer function and the output node uses the linear transfer function.

The Levenberg-Marquardt (LM) algorithm [12] [13] is employed to train the network in this paper. Then the change  $\Delta$  in the weights  $w$  is obtained by

$$\alpha\Delta = -\frac{1}{2}\nabla E \tag{4}$$

where  $E$  is the mean-squared network error

$$E = \frac{1}{M} \sum_{k=1}^M [y(x_k) - d_k]^2 \tag{5}$$

where  $E$  is the mean-squared network error,  $M$  is the number of examples,  $y(x_k)$  is the network output corresponding to the example  $x_k$ , and  $d_k$  is the expected output.

The elements of the  $\alpha$  matrix are given by

$$\alpha_{ij} = (1 + \lambda\delta_{ij}) \sum_{r=1}^m \sum_{k=1}^M \left[ \frac{\partial y_r(x_k)}{\partial w_i} \frac{\partial y_r(x_k)}{\partial w_j} \right] \tag{6}$$

Starting from initial random weights, both  $\alpha$  and  $\nabla E$  are evaluated, and the weight corrections of the network  $w' = w + \Delta$  are obtained by solving Eq.(5). This is known as an LM learning cycle. Each iteration reduces the error until the desired goal is achieved or a minimum is found. The  $\lambda$  in Eq.(6) is adjusted at each cycle according to the error evolution.

The MLP for image restoration is shown in Fig.1. The MLP has the capacity to realize the mapping relationship from a subspace of  $R^n$  to a subspace of  $R^m$  using the training samples. Our proposed method does make use of this characteristic to establish the mapping relationship  $\phi$  between degraded image  $g$  and original image  $\hat{f}$  by means of training samples without a prior knowledge of PSF.

$$\hat{f}_k = \phi(g_k) \quad k = 1, 2, \dots, n \tag{7}$$

The grey levels of the neighborhood around one pixel have some paramount effects on its change when blur occurs. In other words, pixels with the same grey level in an image will probably have different values after the blurring process, if there are different grey levels in their neighborhood. Therefore, it is difficult or even impossible to get an acceptable restoration performance with the method of point-to-point mapping between the clear image and the blurred one. In order to overcome the

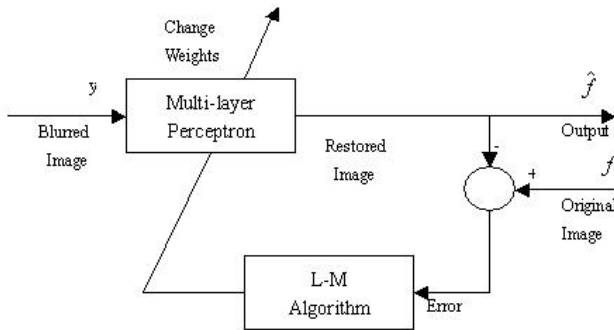


Fig. 1. A MLP model for image restoration

deficiency of point-to-point mapping, we propose a 3×3 sliding-window method to obtain the input vector of the network, which utilizes the region information. The pixel  $A_k$  will have a vector  $p_k$ , termed as Local Neighborhood Vector (LNV), associated with this pixel:

$$P_k = [p_{k1}, p_{k2}, p_{k3}, p_{k4}, p_{k5}, p_{k6}, p_{k7}, p_{k8}, p_{k9}] \tag{8}$$

Given input-output patterns ( $P_k, T_k$ ) as training samples,  $p_k$  is the  $k$  input vector with the form of Eq.(8) and  $T_k$  is the  $k$  desired output vector. The actual network output vector is  $O_k$ . We obtain the network input matrix  $P$  and the target matrix  $T$  (shown in Fig.2, here the size of the image is considered as  $N \times N$  for convenience).

$$\begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & P_{16} & P_{17} & P_{18} & P_{19} \\ \vdots & & & & & & & & \vdots \\ P_{k1} & P_{k2} & P_{k3} & P_{k4} & P_{k5} & P_{k6} & P_{k7} & P_{k8} & P_{k9} \\ \vdots & & & & & & & & \vdots \\ P_{(N-2)^1} & P_{(N-2)^2} & P_{(N-2)^3} & P_{(N-2)^4} & P_{(N-2)^5} & P_{(N-2)^6} & P_{(N-2)^7} & P_{(N-2)^8} & P_{(N-2)^9} \end{bmatrix} \begin{bmatrix} t_1 \\ \vdots \\ t_k \\ \vdots \\ t_{(N-2)^9} \end{bmatrix}$$

Fig. 2. The input matrix  $P$  and target matrix  $T$

Because we implement 3×3 window to extract the input pattern, the index of the last row vector in matrix  $P$  is  $(N-2)^2$ .

Instead of considering all the elements in an image as integrity, the proposed algorithm classifies the training samples into two categories: one is the smoothing region of an image and the other is the region where edges are located. The reason of this separation is that the blurring process has more effects on the part with a larger gradient, compared to the smoothing part. Therefore, we add the edge constraints when we extract the training samples. First we implement edge detection using Sobel operator, then implement network training for the edge region and the smoothing region, respectively.

## 4 Results and Discussion

### 4.1 Network Training

Images are separated into two parts: the training set and the validation set. In order to demonstrate the algorithm fast and conveniently, all the images are set to the identical size  $90 \times 90$  and 256 grey levels. Lenna image (shown in Fig.3) is used as the training set for its abundant texture information. The above algorithm was applied to the synthetic images degraded by a  $9 \times 9$  Gaussian blur with  $\sigma=1$ .

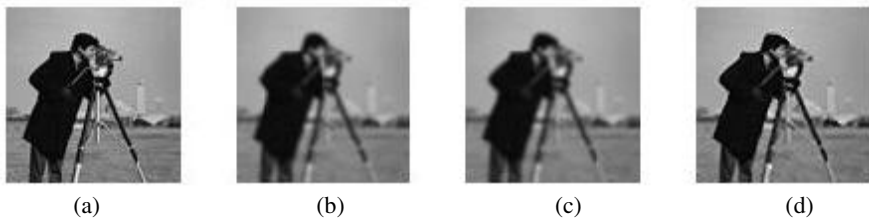


**Fig. 3.** (a) the original Lenna image; (b) Gaussian blurred image

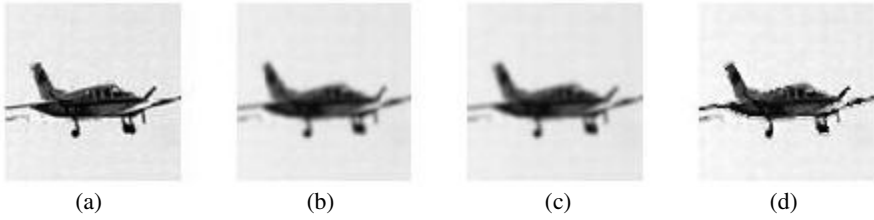
After edge extraction, the input matrix  $P$  and the target matrix  $T$  in the edge region and the corresponding matrix  $P'$ ,  $T'$  in the smoothing region can be gained using the sliding-window method. After inputting the training samples into the MLP designed in Section 3, we will get the trained network parameters, such as the weights matrix and threshold matrix.

### 4.2 Network Validation and Expectation

We choose the blurred image 'cameraman' and 'plane' as validation images. Firstly extract the input matrix of the edge region and the smoothing region matrix. Then input the transformed matrix into the trained neural network in Section 4.1, we will get the corresponding output matrixes  $O$ ,  $O'$  with the same form of  $T$ . Combine  $O$  and  $O'$  into the form of image matrix and then display the image.



**Fig. 4.** Cameraman images (a) original cameraman image; (b) blurred image; (c) restored image with Wiener Filter (d) restored image using proposed method



**Fig. 5.** Plane images (a) original plane image; (b) blurred image; (c) restored image with Wiener Filter (d) restored image using proposed method

### 4.3 Comparison of Restoration Performance

Comparing the restoration performance needs some quality measures that are difficult to define owing to the lack of knowledge about human visual system. We believe that the human objective evaluation is the best ultimate judgment [14]. The performance of image restoration was quantitatively evaluated by normalized mean-square error (NMSE) and improved signal-noise ratio (ISNR) [15], for the original image can be obtained.

$$NMSE = \frac{\|X - \hat{X}\|^2}{\|X\|^2} \quad (9)$$

where  $\hat{X}$  denotes the restored image,  $X$  denotes the original image.

$$ISNR = 10 \log_{10} \frac{\|Y - \hat{X}\|^2}{\|X - \hat{X}\|^2} \quad (10)$$

Where  $Y$  denotes the blurred image;  $\hat{X}$  denotes the restored image;  $X$  denotes the original image. Compute the NMSE and ISNR in terms of formulas (9) and (10). The comparison results are shown in Table 1 and Table 2, respectively.

**Table 1.** NMSE of restored images using Wiener filter and proposed algorithm

NMSE	cameraman	plane
Wiener	0.1003	0.0074
BP NN	0.0039	0.0023

**Table 2.** ISNR of restored images using Wiener filter and proposed algorithm

ISNR	cameraman	plane
Wiener	-9.0753	-2.6221
BP NN	5.0205	2.4147

From Fig.4 and Fig.5, we notice that the details of the restored image with the proposed method is greatly improved, compared to the degraded image and the restored image with Wiener Filter, although it is still not as clear as the original one. From the comparison results in table 1 and table 2, the proposed algorithm is superior to the wiener filter. We repeated all the experiments by using other different degraded images for validation and the similar results were obtained (data not shown). Thus we deduce that Wiener filter usually leads to ring effect and needs the prior knowledge about the PSF, which makes the restoration performance worse, and the proposed method, based on learning mechanism, can effectively demonstrate the inherent relationship between the blurred image and the clear one, which results in better performance.

## 5 Conclusion

In this paper, we formulate the image restoration task as a nonlinear approximation problem in high dimensional space. Region and edge information extracted from the degraded images is used to improve the quality of restoration and accelerate the algorithm speed. The proposed technique differs from the existing ones, which did not use the edge information in modeling. The proposed method uses a sliding window to obtain the network input and develops a good mapping relationship between the blurred image and the target one during the network training process. The performance of the proposed method was evaluated by comparing its results to those of the Wiener filter with the same image. The results show that the method had better performance than the Wiener filter. In the future work, the degraded image with additive noise and more complicated blur like space variant blur will be considered.

**Acknowledgements.** This work was supported in part by China Postdoctoral Science Foundation (No. 2005037503) and by the Science Foundation of Shanghai Municipal Education Commission (No. 05NZ20).

## References

1. Chan, T.F., Shen, J. (ed.): Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods. SIAM Publisher, Philadelphia (2005)
2. Slepian, D.: Linear Least Squares Filtering of Distorted Images. *J. Opt. Soc. Amer.* 57 (1967) 918-922
3. Kundur, D. Hatzinakos, D.: Blind Image Deconvolution. *IEEE Signal Processing Magazine* 13 (3) (1996) 43-64
4. Chen, Y.W., Enokura, T., Nakao, Z.: A Fast Image Restoration Algorithm Based on Simulated Annealing. 3rd Int. Conf. Knowledge-Based Intelligent Information Engineering Systems (1999) 341-344
5. Ayers, G.R., Dainty, J.C.: Iterative Blind Deconvolution Method and Its Applications. *Optics Letters* 13 (7) (1988) 547-549
6. Wen, X., Zhou, L., Wang, D. (ed.): Neural Network Application Design by MATLAB. Science Press, Beijing (2001)

7. Paik, J.K., Katsaggelos, A.K.: Image Restoration Using a Modified Hopfield Network. *IEEE Trans. Image Processing* 1 (1) (1992) 49-63
8. Wong, H.S., Guan, L.: A Neural Learning Approach for Adaptive Image Restoration Using a Fuzzy Model-based Network Architecture. *IEEE Trans. Neural Networks* 11 (3) (2001) 516-531
9. Marinai, S., Gori, M., Soda, G.: Artificial Neural Networks for Document Analysis and Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 27 (1) (2005) 23-35
10. Rizvi, S.A., Wang, L.C., Nasrabadi, N.M.: Nonlinear Vector Prediction Using Feed-forward Neural Networks. *IEEE Trans. Image Processing* 6 (10) (1997) 1431-1436
11. Shang, G., Zhong, L., Chen, L.: Discussion about BP Neural Network Structure and Choice of Samples Training Parameter. *J. Wuhan University of Technology* 19 (2) (1997) 108-110
12. Lampton, M.: Damping-Undamping Strategies for the Levenberg-Marquardt Nonlinear Least-Squares Method. *Computers in Physics* 11 (1) (1997) 110-115
13. Lera, G., Pinzolas, M.: Neighborhood Based Levenberg-Marquardt Algorithm for Neural Network Training. *IEEE Trans. Neural Networks* 13 (5) (2002) 1200 – 1203
14. Zhou, Y.T., Chellappa, R., Vaid, A., Jenkins, B.K.: Image Restoration Using a Neural Network. *IEEE Trans. Acoustics, Speech, and Signal Processing* 36 (7) (1988) 1141-1151
15. Zhang, H.: Image Restoration of Solar Space CCD Using Neural Network. Master thesis, University of Science and Technology Beijing (2002)

# Reflective Symmetry Detection Based on Parallel Projection

Ju-Whan Song<sup>1</sup> and Ou-Bong Gwun<sup>2</sup>

<sup>1</sup> School of Liberal Art, Jeonju University,  
Jeonju, Jeonbuk, Korea  
jwsong@jj.ac.kr

<sup>2</sup> Division of Electronics and Information Engineering,  
Chonbuk National University, Jeonju, Jeonbuk, Korea  
obgwun@chonbuk.ac.kr

**Abstract.** Reflective symmetry is useful for various areas such as computer vision, medical imaging, and 3D model retrieval system. This paper presents an intuitive reflective symmetry detection method for 3D polygon objects. Without any mapping process the method detects the reflective symmetry plane by parallel projection. This paper defines a continuous measure to estimate how much an object is reflective symmetrical for a projection plane through the center of the object. Also it explores the method to detect the reflective symmetry plane with the measure. The proposed method can detect up to 99% reflective symmetry plane not exceeding 4 degree angle for perfect symmetry objects and detect up to 85% reflective symmetry plane not exceeding 10 degree angle for near symmetry objects using Princeton Shape Benchmark.

## 1 Introduction

Now that the image created by 3D computer graphics becomes common in multimedia contents people see them everywhere like games, and web sites. Many studies for 3D model retrieval system actively carry out over the world to use 3D model data on the Web. Though there are many attributes used as feature vectors in 3D model retrieval system, one of them is the reflective symmetry axis.

Symmetry is the native features that many objects in the world have. A human being feels safety and harmony with the symmetrical objects. Reflective symmetry allows us to construct 3D model easily also. Symmetry can be defined as a transformation in  $n$ -dimensional Euclidean space  $E^n$ . Formally, if  $T(s) = T$ , a subset  $S$  of  $E^n$  is symmetric with respect to a transformation. There are three types of symmetric; reflective symmetry, rotational symmetry, translational symmetry. In this paper, we treat only reflective symmetry. Reflective symmetry has a reflection plane or axis, for which the left half-space is a mirror image of the right-half space.

Most symmetry detection algorithms have been devoted to 2D problem [1,2,3]. Identifying symmetries for 3D models is much more complex, so few researches

are performed. The researches for identifying symmetries of 3D models are classified into 2 types. One type is to decide whether an object is symmetric or not, which considers symmetry a binary feature. The other type is to evaluate how much an object is symmetric, which considers symmetry a continuous feature. The continuous symmetry detection algorithms are more practical than the binary symmetry detection algorithms because the continuous algorithms can be used for measuring the symmetry of near symmetry objects.

The binary symmetry detection algorithms are proposed by Wolter et al. [4] and Jiang et al. [5]. Wolter et al. explored a binary algorithm for detecting all rotational symmetries in point sets, polygons, and polyhedra. They transformed the point sets of solids into strings and decided whether it is symmetric or not by pattern matching algorithm. Jiang et al. present a binary symmetry identification algorithm for rotational symmetry based on a scheme called generate and test. They transformed the polyhedral objects into graph representation and decided whether it was symmetrical or not by graph isomorphism. The two algorithms above are dependent on the topology of the object mesh and sensitive to the object geometry.

The continuous symmetry detection algorithms were proposed by Sun et al. [6], Minovic et al. [7], and Kazhdan et al. [8]. Sun et al. converted the symmetry detection problem into extended Gaussian image and identified the symmetry by examining the correlation of the histogram of the Gaussian image. He used perfect symmetric wire-frame objects in order to test his algorithm. Minovic et al. described an algorithm which transformed objects represented by Brep into octree structure and identified symmetries of the polygon objects through an octree traversal. But the octree representation had degenerate cases where it couldn't detect reflective symmetry for regular solids like cubes. Kazhdan et al. presented a measure that evaluated the symmetry of an arbitrary 3D model for all planes through the model's center of mass. It carried on a voxel grid, and was very costly for accurate results.

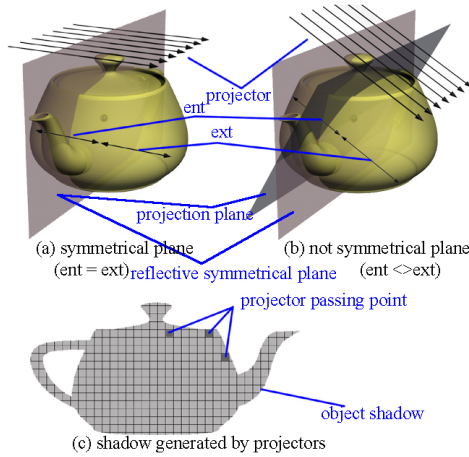
In this paper, we propose a method for identifying the reflective symmetry plane of the 3D model, which is used for a retrieval system on the Web. We assume that the 3D model is constructed by polygon mesh model.

The rest of this paper is structured as follows. Sect. 2 defines a continuous measure to estimate how much an object is reflective symmetrical. Sect. 3 explores the method to detect the reflective symmetry plane with the measure. Sect. 4 evaluates the proposed method with Princeton Shape Benchmark. Finally Sect. 5 concludes by summarizing our work.

## 2 Reflective Symmetry Distance Through Parallel Projector

Figure 1 illustrates our approach. In Fig. 1(a)(b), the planes in the pots are the projection planes and the sets of arrows are projectors. The projection plane passes the object's center of mass. Projectors of Fig. 1(a), pass through the reflective symmetry plane, but those of Fig. 1(b) don't pass through a





**Fig. 1.** Symmetric Disance based on Parallel Projection

reflective symmetry plane. Namely the projection plane of Fig. 1(a) is same to the reflective symmetry plane, but it of Fig. 1(b) is not same to the reflective symmetry plane. Let us denote that the distance from the point which a projector enters into to a point on the projection plane is  $ent_{ij}$  and the distance from the point on the projection plane to a point which a projector exits from is  $ext_{ij}$ . In case when the projection plane is same to the reflective symmetry plane,  $ent_{ij}$  equals  $ext_{ij}$ . But when the projection plane is not the reflective symmetry plane,  $ent_{ij}$  does not equal  $ext_{ij}$ . This property allows us to define the measure SDP(Symmetry Disance based on Parallel projection) which evaluates reflective symmetry distance of 3D objects as follows.

$$SDP(N_x, N_y, N_z) = 1 - \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \frac{|ent_{ij} - ext_{ij}|}{max(ent_{ij}, ext_{ij})} \tag{1}$$

where  $N_x, N_y, N_z$  are the normal vector of a projection plane,  $m$  is the average of the number of projectors composed of the shadow to horizontal direction,  $n$  is the average of the number of it to vertical direction, and  $mn$  is the number of projector composed of the shadow of an object.

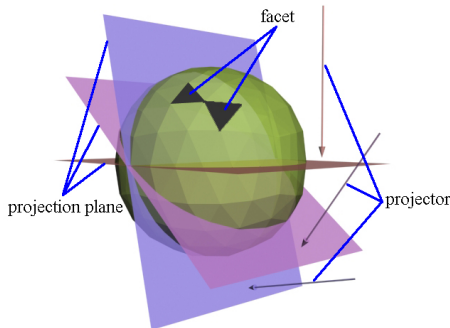
The value of the measure SDP ranges from 0.0 to 1.0. As an object becomes more symmetrical, SDP of the object becomes closer 1.0.

### 3 Detection of Reflective Symmetry Plane

#### 3.1 Detection of Reflective Symmetry Using Parallel Projector

This section describes how to detect symmetry plane by using only parallel projector. First let us basic method from here on. We set the projector planes

passing through the object's center of the mass for all the direction as shown in Fig. 2. And find the value of SDP for the projector planes respectively. The projector plane with the largest value of SDP among them becomes the reflective symmetrical plane of the object. This suggests the following basic method for identifying reflective symmetrical plane.



**Fig. 2.** Detection of Reflective Symmetry using Parallel Projection

```

BasicMethod() {
    Subdivide the surface of a unit sphere into the smaller facets
        of same size by the tessellation algorithm;
    for(each facet){
        Fire a projector from the center of the facet to the center
            of the unit sphere;
        Set the projection plane perpendicular to projector
            for all the facet above respectively;
        Parallel-project the object on the each projection plane
            and obtain the value of SDP;
    }
    Choose the projection plane with the largest value of SDP
        as the reflective symmetry plane;
}
    
```

The basic method above is computationally expensive due to the following reasons.

1. It needs too many subdivisions to detect the exact reflective symmetry plane.
2. The computational cost becomes more expensive in proportion to the product of the number of projector by the number of subdivision.

If the surface of the sphere are subdivided into 20 smaller facets(1 level subdivision), the angle of the detected reflective plane is about 21 degree different from the exact reflective plane. If the surface of the sphere are subdivided into 5120

smaller facets(4 level subdivision), the angle of the detected reflective plane is almost 1 degree angle different from the exact reflective plane. At this time, if the shadow of the object consists of average 100 projectors, 256,000 projectors must be treated. For high resolution subdivision like the case, this method is too slow. The method to reduce the candidate symmetric planes is necessary.

### 3.2 Practical Method by PCA

In the basic method, the search space for reflective symmetry is too large. We reduce the search space, and make the basic method practical using PCA(Principal Component Analysis) and the theorem that any symmetry plane of an object is perpendicular to a principal axis. We find the principal axis using PCA and set the 5 projection planes associated with 5 facets around the principal axis to candidate symmetry plane. Then, we select one among 6 candidate symmetry plane as the detected symmetry plane. For the process, we set a symmetry threshold which is used for deciding whether a projection plane is the reflective symmetry plane or not. Symmetry threshold is tuned depended on the property of the corresponding objects. Our method is described below.

```

PracticalMethod() {
  Obtain the three eigenvalues using PCA;
  if(all eigenvalues is different)
    Find the projection planes perpendicular to 3 axes
      obtained with PCA;
  else if(two engenvalues is equal)
    Find the projection planes perpendicular to 2 axes
      obtained with PCA;
  else
    Find the projection plane perpendicular to 1 axes
      obtained with PCA;
  Compute the value of PCA with parallel-projecting
    the object on each projection planes;
  Select the plane having the largest value of PCA
    as a candidate reflective symmetry plane;
  Find the 6 facets around the facet associated
    the candidate reflective symmetry plane;
  Calculate the PCA of 6 projection planes associated
    with 6 facets;
  if(the largest value of PCA < threshold)
    Process the problem by our basic method(Section 3.1);
  else
    Set the projection plane associated with the largest value
      of PCA as a reflective symmetry plane;
}

```

## 4 Experiments and Evaluations

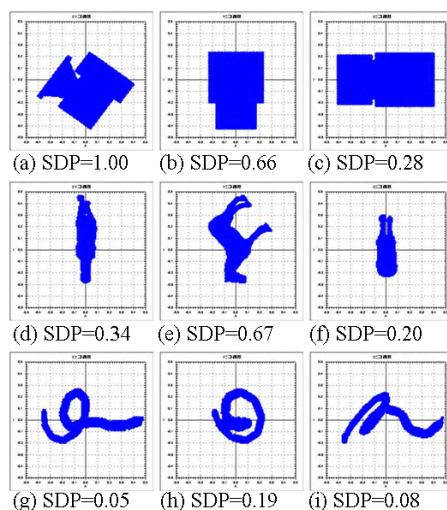
In order to evaluate the proposed method, we implemented the proposed method on a Windows PC with a Pentium 4 CPU(3.2GHZ) and 1GB main memory by Visual C#.NET. We selected 600 out of PSB(Princeton Shape Benchmark) 1814 models and used them as our benchmark data. The models consists of from 16 to 316,000 polygons. The number of their average polygon is about 7,600 polygons. The selected 600 models are classified into 3 types: perfect symmetry(302), near symmetry(264), and not symmetry(34).

The evaluation is done as follows.

1. Does SDP measure the extend of the reflective symmetry of a 3D model?

We selected 3 objects; computer monitor(perfect symmetry), horse(near symmetry), and snake(not symmetry) from the PSB and projected the objects(3D model) on 3 principal projection plane respectively. Fig. 3 shows the results. The projection planes are perpendicular to their 3 principal axes. When the perfect symmetrical object like computer monitor is projected on the projection plane(reflective symmetry plane), the value of SDP is 1.0. When the near symmetrical object like horse is projected on the projection plane(reflective symmetry plane), the value of SDP is 0.67. When not symmetrical object like snake is projected on a projection plane, the value of SDM ranges 0.05-0.19. The value is very small.

Table 1 shows the maximum values of SDP of 600 PSB 3D models. The values of SDP of perfect, near, and not symmetrical objects range from 0.9-1.0, 0.3-1.0, 0.1-0.8 respectively, and its average is 0.96, 0.77, and 0.40 respectively. It shows that the SDP represents the extent of reflective symmetry of 3D models.



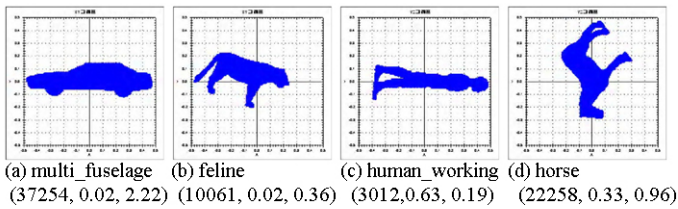
**Fig. 3.** Value of SDP for 3 PSB 3D Models; computer\_monitor(up), horse(middle), snake(down)

**Table 1.** SDP Range of 600 PSB 3D models

SDP	perfect symmetry (number)	near symmetry (number)	not symmetry (number)	total (number)
0.9 ~ 1.0	302	9	0	311
0.8 ~ 0.9	0	131	0	131
0.7 ~ 0.8	0	61	3	64
0.6 ~ 0.7	0	36	3	39
0.5 ~ 0.6	0	19	5	24
0.4 ~ 0.5	0	7	5	12
0.3 ~ 0.4	0	1	7	8
0.2 ~ 0.3	0	0	6	6
0.1 ~ 0.2	0	0	5	5
0.0 ~ 0.1	0	0	0	0
total (number)	302	264	34	600

2. How precisely does the practical method detect the reflective symmetry planes of 3D models ?

Fig.4 shows the typical reflective symmetry planes detected by the practical method. In Fig.4, the symmetry planes are perpendicular to the projection planes. The number in the parenthesis stands for the number of polygon, SDP, and processing respectively. The practical method detected the reflective symmetry planes of perfect symmetry objects(multi\_fuselage, feline) and near symmetry objects(human\_working, horse).



**Fig. 4.** Detected Reflective Plane by Practical Method(the number of polygon, SDP, Processing Time)

Table 2 shows that the proposed method can detect the reflective symmetry plane comparing the detected symmetry plane with the true symmetry plane estimated manually. The practical method can detect up to 99% reflective symmetry plane not exceeding 4 degree angle for perfect symmetry objects, but detect up to 85% reflective symmetry plane not exceeding 10 degree angle for near symmetry objects. In these experiments, the true symmetry plane of near symmetry object must be estimated because it could not be obtained with only the corresponding 3D model data.

**Table 2.** Comparison Of The Detected Symmetry Plane with the True Symmetry Plane

difference (degree angle)	perfect symmetry number(rate %)	near symmetry number(rate %)
0 ~ 2	295(98%)	155(59%)
2 ~ 4	3(1%)	37(14%)
4 ~ 6	2(1%)	8(3%)
6 ~ 8	1(0%)	15(6%)
8 ~ 10	1(0%)	9(3%)
10 ~	0(0%)	40(15%)
models (number)	302(100%)	264(100%)

- How quickly does the practical method detect the reflective symmetry of a 3D model?

We subdivide the surface of a unit sphere into 5120 facets of same size repeating the division 4 times and take the symmetry detection time for the basic method. The basic method takes average 1720 seconds, but the practical method takes 0.51 seconds. Fig. 4 shows the detection time of the reflection symmetry plane for some 3D models. it takes 2.22, 0.36, 0.19, 0.96 seconds for multi\_fuselage, feline, human\_working, and horse respectively.

## 5 Conclusion

In this paper, we presented a novel method which detects reflective plane for polygon mesh model using parallel projection. For the process, we devised an continuous symmetry distance measure which evaluates how much an object is symmetric and made a basic symmetry detection method with the proposed measure. Then, we improved the basic symmetry detection method with PCA. Finally, we identified that the proposed method can detect the reflective symmetry plane. The proposed method found 95% reflective symmetry plane for PSB objects having symmetry(perfect symmetry, near symmetry).

The proposed algorithm is intuitive, understandable, and easy to implement. We are planing to apply the proposed method to 3D object retrieval system for the Web as a forward work.

## References

- Atallah, M. J.: On Symmetry Detection, IEEE Trans. Comput. **C-34** (1985) 663-666
- Sun, C., Si, D.: Fast Reflectional Symmetry Detection Using Orientation Histograms, Real-Time Imaging, **5** (1999) 63-74
- Prasad, V. S. N., Yegnanarayana, B.: Finding Axes of Symmetry From Potential Fields, IEEE Trans. Image Precess. **5** (2004) 1559-1566
- Wolter, J. D., Woo, T. C., Volz, R. A.: Optimal Algorithms for Symmetry Detection in Two and Three Dimensions, The Visual Computer. **1** (1985) 37-48

5. Jiang, X. Y., Bunke, H.: Determination of the Symmetries of Polyhedra and an Application to Object Recognition, LNCS, **553** (1991) 113–121
6. Sun, C., Sherrah, J.: 3D Symmetry Detection Using the Extended Gaussian Image, IEEE Trans. PAMI, **19** (1997) 164–169
7. Minovic, P., Ishikawa, S. Kato, K.: Symmetry Identification of a 3-D Object Represented by Octree, IEEE Trans. PAMI, **15** (1993) 507–514
8. Kazhdan, M. M., Funkhouser, T. A., Rusinkiewicz, S.: Symmetry Descriptors and 3D Shape Matching, Proceedings of the 2004 Eurographics/SGP'04. Eurographics Association, Aire-la-Ville, Switzerland.

# Detail-Preserving Regularization Based Removal of Impulse Noise from Highly Corrupted Images

Bogdan Kwolek

Rzeszów University of Technology  
Computer and Control Engineering Chair  
W. Pola 2, 35-959 Rzeszów, Poland  
bkwolek@prz.rzeszow.pl

**Abstract.** This paper proposes a new filtering scheme for eliminating random-valued impulse noise from gray images. In the first phase a noise detector is utilized to extract the noise candidates. Next, the algorithm applies a connected component analysis in order to gather the neighboring noisy pixels into separate sets of connected noise candidates. The corrupted pixels are restored using a detail preserving regularization method. The main idea of the proposed approach is to gather the noisy candidate pixels into separate sets of connected pixels and solve the minimization functional over these pixels. Experimental results illustrate the efficiency and effectiveness of the algorithm.

## 1 Introduction

Impulse noise can corrupt images due to noisy sensors or channel transmission errors. Typical median filters, which are usually utilized invariantly across the whole images to remove noise, tend to modify both noise pixels and undisturbed pixels. To achieve a good compromise between the image-detail preservation and the noise reduction an impulse detector can be utilized prior to filtering [1]. The filtering is then selectively applied to regions where there is impulse noise. In such decision-based filters the possible noise pixels are first detected and then replaced through a median filter, while all other pixels are unchanged. The adaptive center-weighted median filter (ACWMF) [2] can effectively discover the noise even when its ratio is high. The main drawback is that each noisy pixel is replaced by a median value of neighboring pixels without considering the local structure of the image. The replacement of the noisy pixels by the median involves blurring of edges, which is evidently visible when the noise ratio is high. A recently proposed detail-preserving variational method [3] [4] first detects noisy pixels and then uses a non-smooth data filtering term along with edge preserving regularization to restore the corrupted pixels. The minimization of a convex functional is conducted on the set consisting of all noisy pixels [4]. The nonlinear equation is solved by Newton's method with a suitable initial guess [5].



Our approach detects noisy pixels and additionally applies a connected component analysis in order to gather the neighboring noisy pixels into separate sets of connected noise candidates. To minimize the functional over each set of connected noise candidates we utilize the Levenberg-Marquardt (LM) algorithm. LM can be considered as a combination of steepest descent and the Gauss-Newton method. The steepest descent that is utilized first, guarantees the convergence of the algorithm and the faster Gauss-Newton is utilized finally to achieve the desired tolerance. The ACWMF filter is used to extract noise candidates and its output is utilized as an initial guess for the optimization algorithm. The novelty of our algorithm lies in the use of connected component analysis to gather the noisy candidate pixels into separate sets and to perform a local optimization over these sets. The optimization is then easier. This makes our algorithm several times faster than the algorithm proposed in [3].

The paper is organized as follows. In the next section we briefly review ACWMF filter. In Section 3 we present all ingredients of our method and discuss how our algorithm differs from relevant algorithms. In Section 4 we demonstrate the efficiency and effectiveness of the algorithm using various test images. Some conclusions are drawn in the last section.

## 2 The Adaptive Center-Weighted Median Filter

Let  $x_{i,j}$  be the gray level in a noisy  $M$ -by- $N$  image at pixel location  $(i, j) \in \mathcal{A} \equiv \{1, \dots, M\} \times \{1, \dots, N\}$ . The general expression of the ACWMF filter is as follows:

$$y_{i,j}^{2k} = \text{median}\{x_{i-u,j-v}(2k) \diamond x_{i,j} \mid -h \leq u, v \leq h\}, \quad (1)$$

where  $(2h+1)^2$  is the window size, and  $\diamond$  represents the repetition operation. For  $k = 0, 1, \dots, J-1$ , where  $J = 2h(h+1)$ , we can determine the differences  $d_k = |y_{i,j}^{2k} - x_{i,j}|$ . They satisfy the condition  $d_k \leq d_{k-1}$  for  $k \geq 1$ . To determine if the considered pixel  $(i, j)$  is noisy a set of thresholds  $T_k$  is utilized, where  $T_{k-1} > T_k$  for  $k = 0, 1, \dots, J-1$ . The output of the filter is defined in the following manner:

$$y_{ACWMF} = \begin{cases} y_{i,j}^0, & \text{if } \exists k, d_k > T_k, \\ x_{i,j}, & \text{otherwise,} \end{cases} \quad (2)$$

where  $y_{i,j}^0$  is the output of the standard median filter. For a window of size  $3 \times 3$  four thresholds  $T_k$ ,  $k = 0, \dots, 3$  are needed. Using the median of the absolute deviations from the median  $MAD = \text{median}\{|x_{i-u,j-v} - y_{i,j}^0| : -h \leq u, v \leq h\}$  which is robust estimation of dispersion, we can define the thresholds  $T_k$  as  $T_k = s * MAD + \delta_k$  where  $0 \leq s \leq 0.6$ ,  $\sigma_0 = 40$ ,  $\sigma_1 = 25$ ,  $\sigma_2 = 10$ , and  $\sigma_3 = 5$  [2].

## 3 Our Filter

Our method consists of two steps, which are applied alternatively. The ACWMF filter is utilized to extract noise candidates as well as to provide an initial guess

for the optimization procedure in each iteration  $l = 1, \dots, L$ . Denote by  $\tilde{y}^{(l)}$  the image obtained by applying the ACWMF to the noisy image  $y^{(l-1)}$ . The noise candidate set is extracted through the ACWMF filter and it is extracted on the basis of the following formula:

$$\mathcal{N}_{(l)} = \left\{ (i, j) \in \mathcal{A} : \tilde{y}_{i,j}^{(l)} \neq y_{i,j}^{(l-1)}, \text{ and } y_{i,j}^{(l)} \in \{0, 1, \dots, 255\} \right\}. \quad (3)$$

The set of all uncorrupted pixels in iteration  $l$  is  $\mathcal{N}_{(l)}^C \in \mathcal{A} \setminus \mathcal{N}_{(l)}$  and we keep their original values. Let  $\mathcal{B}_{(l)}$  be a binary image indicating the candidates of noisy pixels. A labeling procedure applied to the image  $\mathcal{B}_{(l)}$  produces the connected components  $\mathcal{C}_{(l)}^{(k)}$ , where  $k = 1, \dots, K$ . Let  $\mathcal{N}_{(l)}^{(k)}$  be a subset of the set  $\mathcal{N}_{(l)}$  whose pixels belong to  $\mathcal{C}_{(l)}^{(k)}$ . Let us now consider a noise candidate at position  $(i, j) \in \mathcal{N}_{(l)}^{(k)}$ . Each of its 4-connected [6] neighbors  $(m, n) \in \mathcal{V}_{i,j}$  is either an undistorted pixel, i.e.  $(m, n) \in \mathcal{N}_{(l)}^C$  or is another noise candidate, i.e.  $(m, n) \in \mathcal{N}_{(l)}^{(k)}$ . The corrupted pixels are then restored by minimizing a convex objective function  $F_{y|\mathcal{N}_{(l)}^{(k)}} : \mathcal{R}^{M \times N} \rightarrow \mathcal{R}$  of the following form:

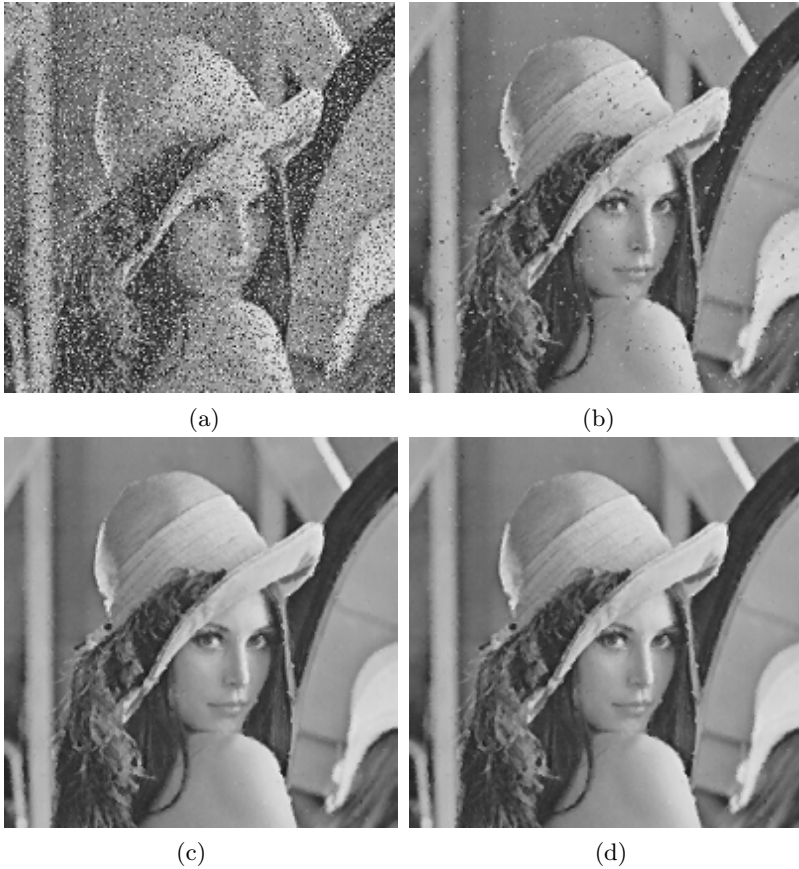
$$\begin{aligned} \mathcal{F}_{y|\mathcal{N}_{(l)}^{(k)}}(\mathbf{u}) &= \sum_{(i,j) \in \mathcal{N}_{(l)}^{(k)}} \left\{ |u_{i,j} - y_{i,j}| + \frac{\beta}{2}(S_1 + S_2) \right\}, \quad (4) \\ S_1 &= \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}_{(l)}^C} 4\phi(u_{i,j} - y_{m,n}), \\ S_2 &= \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}_{(l)}^{(k)}} \phi(u_{i,j} - u_{m,n}), \end{aligned}$$

where  $\beta$  is a regularization factor,  $\phi$  is an edge preserving potential function [7][8]. Examples of such functions are:  $\phi(t) = \sqrt{\alpha + t^2}$  where  $\alpha > 0$  and  $\phi(t) = |t|^\alpha$ ,  $1 < \alpha \leq 2$ . In the output image  $y^{(l)}$  the corrupted pixels are set to values generated by the optimization procedure, whereas all undistorted pixels are copied from the  $y^{(l-1)}$ . The data-fitting term  $|u_{i,j} - y_{i,j}|$  prevents the wrongly detected undistorted pixels from being modified to other values, whereas the regularization term  $(S_1 + S_2)$  accomplishes the edge-preserving smoothing of corrupted pixels [3][4]. The regularization factor balances the effects of the data-fitting term and the *a priori* term. In our approach the noise candidates are restored by minimizing the functionals  $\mathcal{F}_{y|\mathcal{N}_{(l)}^{(k)}}(\mathbf{u})$ ,  $k = 1, \dots, K$ , whereas [3][4] restore the noise candidates by minimizing a single functional that is restricted to the noise candidate set  $\mathcal{N}_{(l)}$ .

### 4 Tests

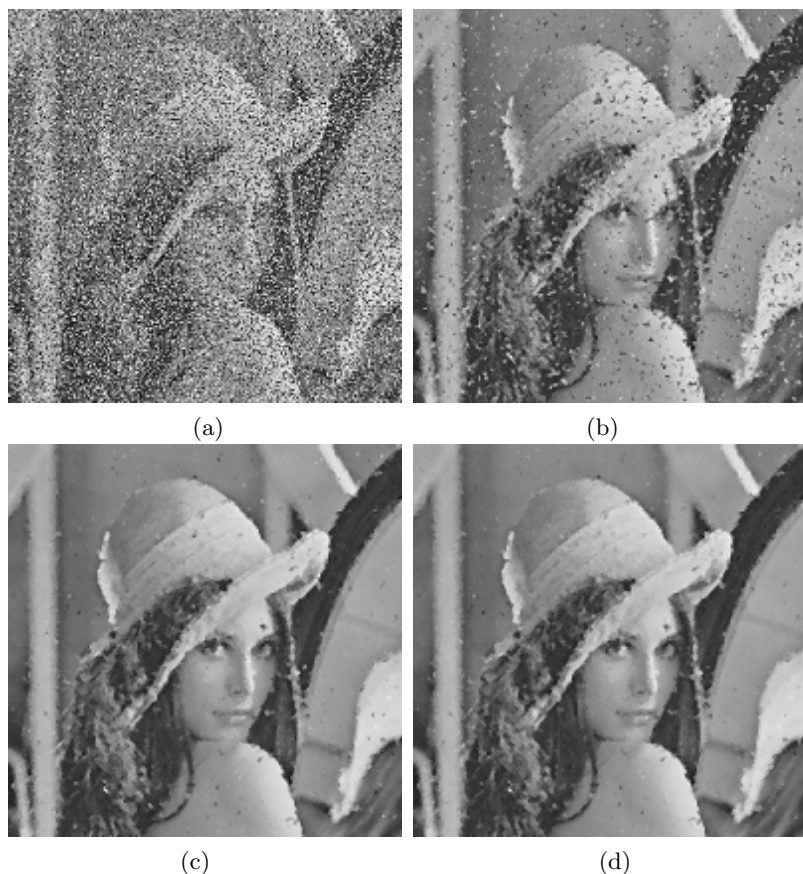
In this section we compare our method with ACWMF [2] and detail-preserving regularization [3] in terms of restoration errors and computation time. In all

images, 30% or 50% pixels were corrupted with random-valued impulse noise, see Fig. 1. and Fig. 2. The peak signal to noise ratio (PSNR) and mean absolute error (MAE) [9] have been utilized to measure restoration errors. The potential function  $\phi(t) = |t|^{1.3}$  has been applied in all experiments.



**Fig. 1.** Image with 30% noise (a). Restored images by ACWMF with  $s = 0.3$  (b), our method in 3 iterations (c), and in 4 iterations (d) with  $\beta = 3.0$ ,  $s = 0.6$  in 1-st it.,  $s = 0.5$  in 2-nd it., and  $s = 0.2$  in the next iterations.

The results from Table 1-2 indicate that errors obtained by our method in 3 iterations are quite comparable with errors that have been obtained by method [3] in four iterations. The method is several times faster than the mentioned above method, see Table 3 where computation time of the LM procedure is related to processing time of ACWMF. The algorithms were implemented in C and run on a PC workstation with a Pentium IV 2.4 GHz processor. The work [3] reports that for 30% noise the minimization procedure takes 30 times more CPU



**Fig. 2.** Image with 50% noise (a). Restored images by ACWMF with  $s = 0.1$  (b), our method in 3 iterations (c), and in 4 iterations (d) with  $\beta = 3.0$ ,  $s = 0.6$  in 1-st iteration and  $s = 0.2$  in the next iterations.

time than ACWMF. In our approach the optimization procedure takes about 3 times more CPU time than ACWMF. In comparison with ACWMF the proposed algorithm yields superior subjective quality with respect to impulse noise cancellation and image detail preservation. The SolvOpt optimization procedure [10] that allows for minimization nonlinear, possibly non-smooth nonlinear functions has also been tested in our algorithm. However, the computation time of this procedure is far longer than processing time of LM.

In order to test how good the noise cancellation is we performed an optimization-based restoration of noisy images assuming that the noise detector is perfect. The LM procedure employing such perfect noise indicator and operating on connected noise candidates restores in one iteration the image *lena* corrupted by 30% and 50% impulse noise with PSNR=32.8 dB and PSNR=29.0 dB, respectively.

**Table 1.** Restoration errors at 30% noise

		<i>bridge</i>	<i>camera</i>	<i>goldhill</i>	<i>lena</i>
PSNR	Noisy image	14.37	13.69	14.37	14.60
	ACWMF	23.82	23.32	25.03	27.03
	Our method	25.27	24.75	27.42	30.16
MAE	Noisy image	22.24	23.42	21.87	21.42
	ACWMF	6.47	5.06	4.90	3.35
	Our method	5.92	3.97	4.11	2.31

**Table 2.** Restoration errors at 50% noise

		<i>bridge</i>	<i>camera</i>	<i>goldhill</i>	<i>lena</i>
PSNR	Noisy image	12.00	11.54	12.23	12.40
	ACWMF	19.19	18.11	20.02	20.98
	Our method	22.68	22.26	24.46	25.93
MAE	Noisy image	37.00	38.56	36.03	35.52
	ACWMF	14.11	13.88	11.90	9.80
	Our method	9.77	7.22	7.16	4.99

**Table 3.** Computation time [sec.]

	ACWMF	LM-1st. it.	LM-2nd. it.	LM-3rd. it.
30% noise	0.34	0.81	0.15	0.07
50% noise	0.35	1.04	0.46	0.21

Next, we compared our method with recently proposed techniques. In [11], Luo reports restoration results in PSNR for images corrupted by 30% random-valued impulse noise. For example, for standard image *lena* of size  $256 \times 256$  this work reports the following restoration results: ACWMF - 27.18 dB, iterative procedure [3] - 28.33 dB, algorithm-based on alpha-trimmed mean [11] - 28.48 dB. Taking into account results from Table 1 it is evident that our method provides significant improvement over all other approaches.

## 5 Conclusion

This paper considers the 2-phase methods in removal of impulse noise from highly corrupted images. We propose a new method for eliminating random-valued impulse noise from gray images. The main idea of our approach is to gather the noisy candidate pixels into individual sets of connected pixels and solve the minimization functional over these pixels. To minimize the functional over each set of connected noise candidates we utilize the Levenberg-Marquardt algorithm. The ACWMF filter is used to extract noise candidates and its output is utilized as an initial guess for the optimization algorithm. Our method can speed up the computations and the restored images are better. Experimental

results indicate that the images are restored with satisfactory quality even at very high level of impulse noise. In our experiments with highly corrupted images the proposed algorithm performed better on all test-images than other relevant 2-phase algorithms.

**Acknowledgments.** This work has been supported by Polish Ministry of Education and Science (MNSzW) within the projects 3 T11C 057 30 and N206 019 31/2664.

## References

1. Lukac, R., Smolka, B., Plataniotis, K.N., Venetsanopoulos, A.V.: Angular multi-channel sigma filter. In: IEEE Int. Conf. on Acoustics, Speech, and Signal Processing. (2003) 745–748
2. Chen, T., Wu, H.R.: Adaptive noise detection using center-weighted median filters. IEEE Signal Proc. Letters **8** (2001) 1–3
3. Chan, R., Hu, C., Nikolova, M.: An iterative procedure for removing random-valued impulse noise. IEEE Signal Proc. Letters **11** (2004) 921–924
4. Chan, R., Ho, C., Nikolova, M.: Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. IEEE Trans. Image Proc. **14** (2005) 1479–1485
5. Chan, R., Ho, C.W., Nikolova, M.: Convergence of newton’s method for a minimization problem in impulse noise removal. J. of Comp. Math. **22** (2004) 168–177
6. Haralick, R.M., Shapiro, L.G.: Computer and robot vision. Addison-Wesley (1992)
7. Black, M., Rangarajan, A.: On the unification of line process, outlier rejection, and robust statistics with application to early vision. Int. J. Comput. Vision **19** (1996) 57–91
8. Charbonnier, P., Blanc-Fraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computed imaging. IEEE Trans. Image Proc. **6** (1997) 298–311
9. Bovik, A.: Handbook of image and video processing. Academic Press, San Diego (2000)
10. Kuntsevich, A., Kappel, F.: SolvOpt - The solver for local nonlinear optimization problems. Inst. for Mathematics, Karl-Franzens University of Graz (1997)
11. Luo, W.: An efficient detail-preserving approach for removing impulse noise in images. IEEE Signal Proc. Letters **13** (2006) 413–416

# Fast Algorithm for Order Independent Binary Homotopic Thinning

Marcin Iwanowski<sup>1</sup> and Pierre Soille<sup>2</sup>

<sup>1</sup> Institute of Control and Industrial Electronics, Warsaw University of Technology  
ul.Koszykowa 75, 00-662 Warszawa, Poland  
iwanowski@isep.pw.edu.pl

<sup>2</sup> Institute for Environment and Sustainability  
DG Joint Research Centre, European Commission  
T.P. 262, I-21020 Ispra (VA), Italy  
Pierre.Soille@jrc.it

**Abstract.** In this paper an efficient queue-based algorithm for order independent homotopic thinning is proposed. This generic algorithm can be applied to various thinning versions: homotopic marking, anchored skeletonisation, and the computation of the skeleton of influence zones based on local pixel characterisations. An example application of the proposed method to detect the medial axis of wide river networks from satellite imagery is also presented.

## 1 Introduction

Homotopic thinning algorithms can be used to skeletonise discrete binary patterns for a wide variety of pattern recognition tasks as well as for extracting the medial axis of thick and elongated objects before vectorisation. However, most algorithms produce results which are order dependent in the sense that the output skeleton depends on some arbitrary choice such as the order in which the image pixels are processed. We present an efficient algorithm based on queue data structures able to compute order independent anchor skeletons on large data sets. The conditions for order independent removable pixels have been proposed in [1].

Algorithm for three types of thinning are described in the paper. The first one deals with homotopic marking where removable pixels are those pixels which can be removed without modifying the homotopy of the input image. These pixels are called simple pixels. The second type of thinning is obtained when the notion of homotopy that is the basis of the definition of simple pixels, is replaced by the notion of background homotopy. This notion leads to a different kind of simple pixel which we call the b-simple pixel. Thinning with b-simple pixels leads to a skeleton of influence zones based on local pixel characterisations (which will be called local-SKIZ in the paper). Finally, anchored skeletonisation allows the user to define a set of pixels which cannot be removed during the thinning process: the anchor pixels. The appropriate choice of anchor pixels (e.g., the local or regional maxima of a distance function), results in a skeleton which better models the

input shape. Thanks to the application of queue data structures, the computation time of the proposed algorithm is much shorter than the classic algorithm with multiple scans of the whole image.

The paper is organised as follows. Section 2 contains the necessary background notions. A fast queue-based algorithm is developed in section 3. Section 4 illustrates the use of the algorithm and section 5 concludes the paper.

## 2 Background Notions

### 2.1 Simple and B-Simple Pixels

Usually, different connectivity types are used for foreground and background pixels. If  $\mathcal{G}$  is the foreground connectivity, then  $\mathcal{G}' = 12 - \mathcal{G}$  is the background connectivity. Hence, two combinations are possible for the square grid:  $\mathcal{G} = 8$  (and  $\mathcal{G}' = 4$ ) or  $\mathcal{G} = 4$  (and  $\mathcal{G}' = 8$ ).

Two binary images are background (foreground) homotopic [1] if and only if there exists a one to one correspondence (i.e., bijective mapping) between the elements of a set of connected components of the background (resp. foreground) of both images. Two images are homotopic if and only if they are both foreground and background homotopic.

A pixel  $p$  belonging to the image  $F$  is *simple* [2,3,4] if and only if its removal from the image does not change the homotopy of  $F$ . By replacing the homotopy by background homotopy in this definition we get the definition of a *b-simple* pixel. It was proven in [2,3] that the simpleness of a pixel can be determined by analysing its  $3 \times 3$  neighbourhood. Examples of various configurations of binary pixels are displayed in Fig. 1.

000	100	011	111	001	011
010	010	111	110	010	110
000	000	111	100	010	010
a	b	c	d	e	f

**Fig. 1.** Examples of binary pixel configurations: (a) Isolated for  $\mathcal{G} = 4$  and  $\mathcal{G} = 8$ . (b) Isolated only for  $\mathcal{G} = 4$ . (c) Simple for  $\mathcal{G} = 4$  and inner for  $\mathcal{G} = 8$ . (d) Simple for  $\mathcal{G} = 4$  and  $\mathcal{G} = 8$ . (e) Simple for  $\mathcal{G} = 4$  but not for  $\mathcal{G} = 8$ , (f) Simple for  $\mathcal{G} = 8$  but not for  $\mathcal{G} = 4$ . Every simple pixel as well as every isolated one is b-simple.

When investigating the order independent simple or b-simple pixel, its relation with neighbouring pixels of the same kind must be analysed. In order to check whether a pixel  $p$  is independent simple, we must test its pairwise independence from all its simple neighbours. When simple pixels for  $\mathcal{G} = \{4, 8\}$  are investigated,  $p$  should be simple and independent from  $q$  for  $\mathcal{G} = 4$  and for  $\mathcal{G} = 8$  simultaneously.

Figure 2 shows generic configurations of intersecting neighbourhood of two pixels. To get all possible configurations, one should add configurations that are



*0q	*0q	*1q	*00*	*01*	*10*	*11*	*01*	*01*	*10*	*01*	*10*	*11*
*p0	*p1	*p1	*pq*	*pq*	*pq*	*pq*	*pq*	*pq*	*pq*	*pq*	*pq*	*pq*
***	***	***	*00*	*00*	*00*	*00*	*10*	*01*	*10*	*11*	*11*	*11*
a	b	c	d	e	f	g	h	i	j	k	l	m

**Fig. 2.** Configurations of common 8-neighbours of two 8-adjacent pixels  $p$  and  $q$ , '\*' stands for any pixel value

symmetrical to b,e,f,g,h,k, and l with respect to the  $pq$  axis, as well as rotations through  $90^\circ, 180^\circ$ , and  $270^\circ$ . Configurations of independent pixels [1] from Fig. 2 are given in table 1. Note that some configurations do not exist for  $\mathcal{G} = 4$  and consequently for  $\mathcal{G} = \{4, 8\}$ .

**Table 1.** Independent configurations from Fig. 2

	$\mathcal{G} = 8$	$\mathcal{G} = 4$	$\mathcal{G} = \{4, 8\}$
independent simple	b,c,e,f,g	a,b,g,k,l	b,g
independent b-simple	a,b,c,d,e,f,g	a,b,d,g,k,l	a,b,d,g
non existing	-	e,f,h,i,j	e,f,h,i,j

If the given simple pixel is independent from all of its simple neighbours, the multiple configuration of simple pixels must be checked because isolated triple and quadruple configurations of simple pixels cannot be removed even if they are independent. The last check is not required for b-simpleness case, where isolated triples or quadruples can be removed.

One of the configurations (marked as c in Fig. 2) requires more attention when  $\mathcal{G} = 4$  and  $\mathcal{G}' = 8$ . This is due to the fact that for some configurations of '\*'-neighbours of  $p$  this pixel can be removed, even if it is not independent. The 'edge' pixel is characterised as a pixel having three adjacent simple (resp. b-simple) neighbours, whereby the diagonal neighbour is dependent while two non-diagonal ones are independent. This produces the configuration shown in Fig. 3g.

0000	000	0000	0000	1000	0000	****
0010	010	0110	0110	0110	0p11	*rq*
0100	010	0100	0110	0110	01q1	0ps*
0000	000	0000	0000	0001	0000	*0**
a	b	c	d	e	f	g

**Fig. 3.** Multiple simple neighbours. (a,b) isolated pair of simple pixels. (c) isolated triple of simple pixels. (d) isolated quadruple of simple pixels (for  $\mathcal{G} = 4$  and  $\mathcal{G} = 8$ ). (e) isolated quadruple of simple pixels (for  $\mathcal{G} = 4$  only). (f) example of configuration where  $p$  is dependent of  $q$ , but can be removed ( $\mathcal{G} = 4$ ). (g) generic 'edge' configuration of neighbourhood ( $\mathcal{G} = 4$ , '\*' stands for any pixel values, but such that  $r, q, s$  remain simple).

Since  $p$  is independent of  $r$  and  $s$ , the rest of its 4-adjacent neighbours must be background neighbours. Otherwise  $p$  would not be simple. In addition, in case of simple pixels (not b-simple), the quadruple one being considered must be checked whether it is isolated. If so,  $p$  cannot be removed.

## 2.2 Principal Types of Thinning

**Homotopic Marking.** In the simplest case, during the thinning process, all simple pixels are iteratively removed by setting them to 0. The removal of a simple pixel, by definition, preserves the homotopy of the input image  $F$ . This type of thinning can be (and sometimes is) called skeletonisation. However, since it does not reflect properly the shape of the input figure (as a skeleton is expected to do), it is usually called a *homotopic marking* [5,6,7]. It reduces binary objects to single pixels or to loops surrounding the holes inside the figures (preserving the homotopy). For example, the block letters 'K', 'M' or 'T' are reduced to a single pixel although they are clearly different. Homotopic marking can be thus considered as a skeleton satisfying the topological constraints (preserving the homotopy) but not the geometrical ones [8].

**Skeleton by Influence Zones.** The replacement of the simple pixels by the b-simple pixels in the thinning procedure leads to a (non-homotopic) skeleton preserving only those connected components of the homotopic marking which are not simply connected. Contrary to the previous method, in this one connected components without holes disappear. That is this type of a skeleton resembles the skeleton of influence zones (SKIZ [5,9,6]) of the complement of the input image. Comparing to the distance-based SKIZ the result of thinning with b-simple pixels contains some lines which do not separate different influence zones, but are located inside the same one. This is due to the fact that the actual SKIZ cannot be obtained using local characterisation of pixel neighbourhood [10]. To stress this difference the thinning by removal of b-simple pixels will be called *local-SKIZ* (SKIZ based on local pixel characterisation). In order to get the actual SKIZ (which, contrary to local-SKIZ, is not b-homotopic to the input image) one needs to label the connected components of the background in advance and add one additional test when testing the b-simpleness. According to this test, if a non-b-simple pixel has among its background neighbours all pixels with the same label it can be removed during the thinning process. This approach requires also that when a pixel is removed, the closest background label must be propagated.

## 2.3 Anchored Skeletonisation

In order to get a skeleton that better characterises the input shape, an *anchored skeletonisation* has been proposed [11]. It requires the prior definition of a set of pixels which, by definition, cannot be removed during the thinning process.

These pixels are called anchor pixels and are quite often used in various skeletonisation algorithms [12,11,13,14,7]. They are usually derived from a distance function [15,16,6] and are computed either as local maxima of a distance function<sup>1</sup> or as its regional maxima<sup>2</sup>. The anchored skeleton with regional maxima used as anchor pixels is called the *minimal skeleton*.

### 3 The Algorithm

Thinning requires multiple scans of the image in order to find and remove the appropriate pixels until stability is reached. Since we are proposing a generic algorithm, i.e., applicable to any of the described thinning methods, we now introduce the notion of *removability* which refers to all kinds of simpleness. A removable pixel is defined here as a pixel which can be removed according to a given criterion, i.e., simple, simple but not anchor, or b-simple. There are various approaches to the thinning from an algorithmic point of view. One of the earliest is a *sequential* one. In this approach, when scanning the image, once a removable pixel is found, it is modified before proceeding to the next pixel. The final result strongly depends on the order of processing of the pixels. For example, in the case of homotopic marking, a rectangle is reduced to the lower-right pixel when considering the forward raster scanning order, instead of the upper-left pixel for a backward raster scan. In order to solve this problem, a two-stage iterative process is usually applied. For each iteration, first all removable pixels are initially detected (detection phase) and are subsequently removed (removal phase). Another important issue refers to the pixel scanning sequence. Usually in the detection phase, the entire image is scanned to find removable pixels requiring multiple time-consuming scans. However, the full image scans can be performed only once at the beginning to detect an initial set of removable pixels. Indeed, full scans are not compulsory because new removable pixels can appear only within the neighbourhoods of already modified pixels. This type of algorithm requires an application of a *queue* data structure [11].

#### 3.1 The Main Body of the Algorithm

The fast algorithm makes use of two first-in first-out queues and is presented in Algorithm 1. The input parameters for the algorithm are: input image  $f$ , type of foreground and background connectivity  $\mathcal{G}, \mathcal{G}'$  and thinning method (which drives the *isRemovable* function).

<sup>1</sup> The pixel  $p$  is a local maximum if and only if there is no pixel of higher value than  $p$  in  $N_{\mathcal{G}}(p)$ . The set of local maxima of a distance function is also called a *skeleton by opening*.

<sup>2</sup> The regional maximum is defined as a connected set of pixels of the same value such that all pixels belonging to its external boundary have values strictly lower. The set of regional maxima of a distance function is equivalent to an ultimate eroded set of the input binary image [6].

**Algorithm 1.** Fast queue-based thinning

$f, s$  images: input and temporary image used for flagging purposes  
 $Q_{\text{current}}, Q_{\text{next}}, Q_2$  “first-in first-out” queues

```

1. for each pixel  $p$  do
2.   if  $isRemovable(p) = \text{true}$  then  $\text{add}(p, Q_{\text{current}})$  ;  $s(p) \leftarrow 1$ 
3.   else  $s(p) \leftarrow 0$ 
4. do
5.    $modified \leftarrow \text{false}$ 
6.   while  $\text{empty}(Q_{\text{current}}) = \text{false}$  do
7.      $p \leftarrow \text{retrieve}(Q_{\text{current}})$  ;  $s(p) \leftarrow 0$  ;  $modified \leftarrow \text{true}$  ;  $f(p) \leftarrow 0$  ;  $\text{add}(p, Q_2)$ 
8.   if ( $modified = \text{true}$ ) then
9.     while  $\text{empty}(Q_2) = \text{false}$  do
10.       $p \leftarrow \text{retrieve}(Q_2)$ 
11.      for all  $q \in \mathcal{N}_G(p)$  do
12.        if  $isRemovable(q)$  and  $s(q) = 0$  then  $\text{add}(q, Q_{\text{next}})$  ;  $s(q) \leftarrow 1$ 
13.       $Q_{\text{current}} \leftarrow Q_{\text{next}}$ 
14. while  $modified = \text{true}$ 

```

The algorithm proceeds in two steps. In the first step (lines 1–3), the entire image is scanned while removable pixels are inserted in the queue. In the second step (lines 4–14) pixels are successively retrieved from the queue and removed (set to 0) from image  $f$  in consecutive iterations. The second step proceeds according to an iterative two-phase scheme. The two phases are performed inside the main loop (lines 4–14). The loop is controlled by the Boolean variable  $modified$  which is true if in a given iteration at least one pixel has been removed from the image. The detection and removal phases are performed in two internal loops. The first loop (lines 6–7) is controlled by the content of the queue  $Q_{\text{current}}$  and pixels are removed. Once a pixel has been removed, it is added to queue  $Q_2$ . The second loop (lines 9–10) scans the pixels of  $Q_2$  analysing their neighbours and adding new removable pixels to the queue  $Q_{\text{next}}$  (line 12).

In order to separate pixels removed from the queue and the ones which are added to the queue for further processing in the next iteration, two queue structures are used:  $Q_{\text{current}}$  and  $Q_{\text{next}}$ . Pixels are removed from the first queue, whereas new pixels are added to the second queue. At the end of the loop (line 13), the content of queue  $Q_{\text{next}}$  is copied to the queue  $Q_{\text{current}}$ . In practice, instead of copying the queues, their pointers are exchanged (line 13).

### 3.2 Removability Test

The Boolean value returned by the function  $isRemovable(p)$  (removability test called at lines 2 and 12) depends on the chosen thinning method and assumed connectivity for foreground and background.

The case of **homotopic marking** requires both the simpleness check, and the independence test. The pixel  $p$  is first tested using the simpleness check. If  $p$  is simple, then the pairwise independence test is performed. According to

this test, all simple pixels  $q \in N_{\mathcal{G}}^1(p)$  are analysed and for all pairs  $p, q$  the independence is investigated. When  $p$  is pairwise independent of all  $q$  and  $\mathcal{G} = 4$ , the test is terminated and *isRemovable*( $p$ ) returns **true**. If a positive test results in case of  $\mathcal{G} = 8$ , another test must be performed in order to test the isolated triple and quadruple configurations. If  $p$  is not in any of these configurations, then *isRemovable*( $p$ ) returns **true**. Otherwise (when the result of a pairwise independence test is negative for one of the diagonal neighbours) when  $\mathcal{G} = 4$ , an additional test of 'edge' configuration must be performed. If  $p$  is in such a configuration, but not in the configuration of an isolated quadruple, then  $p$  can be removed despite its pairwise dependence so that **true** is returned by *isRemovable*( $p$ ). In all other cases, the latter function returns **false**.

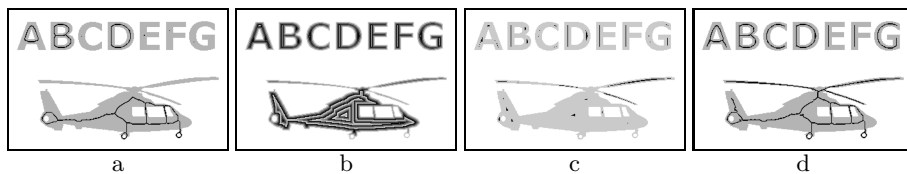
The case of **anchored skeletonisation** proceeds in the same manner as the previous case except that an anchor image must be considered. The only difference is that when pixel  $p$  is tested for simpleness, it cannot be simple if it is an anchor pixel. Also, when considering independence of the neighbours, an adjacent pixel which is an anchor pixel cannot be considered as simple. Thus  $p$  is always independent of all  $q \in N_{\mathcal{G}}^1(p)$  such that  $a(q) = 1$ .

Finally, in **local-SKIZ** case, the pixel  $p$  is first checked using the b-simpleness test. If it is b-simple, then the pairwise independence test is performed. All simple pixels  $q \in N_{\mathcal{G}}^1(p)$  are then analysed and all pairs  $p, q$  are investigated to check whether  $p$  is independent of  $q$  (see table 1, second row). If  $p$  is pairwise independent of all  $q$ , the function *isRemovable*( $p$ ) returns **true**. Otherwise (when  $p$  is not pairwise independent from one of its diagonal neighbours) if  $\mathcal{G} = 4$ , the supplementary test of the 'edge' configuration must be performed. If  $p$  is in such a configuration, then  $p$  can be removed despite its dependence and **true** is returned by *isRemovable*( $p$ ). In all other cases it returns **false**.

Look-up tables can be used to increase the processing speed of simpleness and b-simpleness tests. The number of possible configurations of 3x3 neighbourhood is equal to 256. Thus, every configuration can be coded via 0's and 1's referring to the value of particular neighbours as a binary number, whose decimal equivalent is the index of the table. The value referring to this index is computed in advance according to the definition of a simple or b-simple pixel for a given foreground and background connectivity. The same trick can be applied to pairwise independence test.

## 4 Results

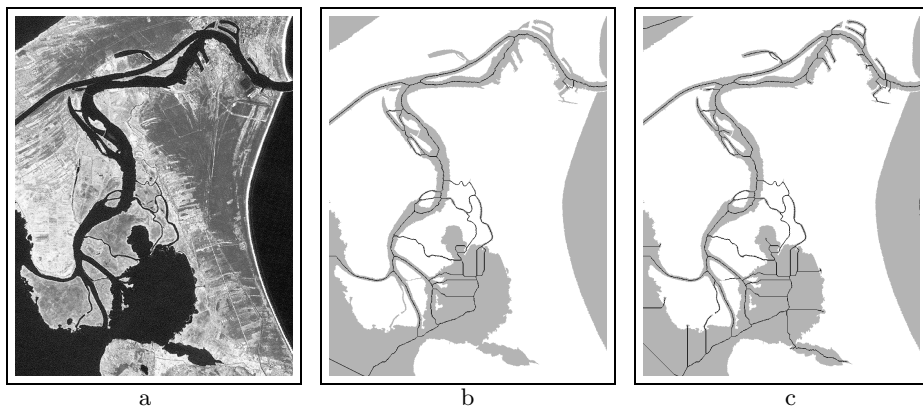
Anchored skeletonisation is illustrated in Fig. 4. The homotopic marking (see Fig. 4a) reflects the homotopy of the original image (marked in grey) but is not preserving the long and thin shape details. To generate a set of anchor pixels, first an 8-connected distance function was computed (see Fig. 4b). Then, the 8-connected regional maxima of this function were extracted (see Fig. 4c) and used as anchor pixels. The resulting anchored skeleton is shown in Fig. 4d. Compared to homotopic marking, the final shape is much better modelled and shows all important elongated details.



**Fig. 4.** Anchored order-independent skeletonisation for  $\mathcal{G} = 8$ : a - skeleton without anchors (homotopic marking), b - 8-connected distance function, c - regional maxima of the distance function, d - anchored skeleton

The method presented was applied to the extraction of wide river networks from a Landsat 7 panchromatic image. The test image of size 972x1024 shows a part of the Odra river outlet close to Swinoujscie (North-Western Poland), where the river flows into the Baltic sea (see Fig. 5a). The input image was pre-processed in order to prepare the input for skeletonisation. During the pre-processing step, first, small variations were filtered out by an opening by reconstruction (size 5 in 8-connected grid). The filtered image was binarised at level 40 and modified using the morphological fill-hole method (in 8-connected grid) in order to remove areas which were disjoint with the river network. A supplementary area opening (of 15 pixel size) was also applied in order to remove small islands which do not have to be considered during the river network detection process.

In order to get the river network from binary image, first, the non-anchored skeletonisation (homotopic marking) was performed (see the result in Fig. 5b). The river and channel network was detected properly as a skeleton line. This network however does not contain the branches corresponding to the dead ends of



**Fig. 5.** Application of the proposed algorithm to the computation of the medial axis of a river network. a Input image, b non-anchored skeleton (homotopic marking), c anchored skeleton. The binary input for skeletonisation is marked in grey, it relates to water areas. North is pointing rightwards.

the channels. In order to detect them, the anchored skeletonisation method was applied. To prepare the set of anchor pixels, the 4-connected distance function was computed. Its 8-connected maxima were used later as the set of anchor pixels for the skeletonisation, the result of which is shown in Fig. 5c. In the latter case, the complete network with dead-ends was detected<sup>3</sup>. Both skeletons were produced using an order-independent skeletonisation with  $\mathcal{G} = 8$  and  $\mathcal{G}' = 4$ .

The efficiency of the proposed algorithm was also tested. As expected, the queue structure shortens the computing time of the thinning algorithm. In the queue-based algorithm, multiple image scanning, i.e., where every pixel is checked for its simpleness, is avoided because only those pixels that could possibly become simple are tested once the queue has been initialised.

The proposed algorithm was compared to the classical iterative approach where in each iteration the entire image was scanned twice, first when simple pixels were detected and second, when they were removed. Two test images were used. The first image was a 400x400 image containing 72 relatively thin 8-connected components (which form 112 4-connected ones). The second test image was a binarised Landsat 7 satellite image which was used in the river network detection example. The image dimensions were 972x1024 pixels and contained a single thick 8-connected component referring to the river network mask (and several tiny ones). This mask is shown on Fig. 5b and c in grey.

Depending on the test image size and complexity, the proposed algorithm was 20 to 70 times faster than the classical one.

## 5 Conclusions

In this paper a generic queue-based implementation of order independent thinning developed in [1] has been presented. It speeds-up the computation of thinning compared to the classical algorithm, because it scans the whole image only once to initialise the queue. The algorithm allows the computation of various types of thinning in different connectivity schemes. Computing time measurements confirm the superiority of the proposed algorithm over the non-queue implementation.

## References

1. Iwanowski, M., Soille, P.: Order independence in binary homotopic thinning. In: Proc. of Discrete Geometry for Computer Imagery'06. (Volume 4245 of Lecture Notes in Computer Science.)
2. Rosenfeld, A.: Connectivity in digital pictures. *Journal of the ACM* **17** (1970) 146–160

---

<sup>3</sup> As it can be seen on the figure, there has been some 'false' dead-ends detected. This was due to some configurations of channel boundary, which resulted in additional maxima of the distance function. This problem can be solved either by pre-processing the original image for distance function computation, or by the filtering of the distance function maxima. This issue is however out of the scope of this paper.

3. Kong, T., Rosenfeld, A.: Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing* **48** (1989) 357–393
4. Kong, T.: On topology preservation in 2-D and 3-D thinning. *International Journal of Pattern Recognition and Artificial Intelligence* **9** (1995) 813–844
5. Serra, J.: *Image analysis and mathematical morphology*. Academic Press, London (1982)
6. Soille, P.: *Morphological Image Analysis: Principles and Applications*. corrected 2nd printing of the 2nd edn. Springer-Verlag, Berlin and New York (2004)
7. Ranwez, V., Soille, P.: Order independent homotopic thinning for binary and grey tone anchored skeletons. *Pattern Recognition Letters* **23** (2002) 687–702
8. Ronse, C.: Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images. *Discrete Applied Mathematics* **21** (1988) 67–79
9. Serra, J., ed.: *Image analysis and mathematical morphology. Volume 2: Theoretical advances*. Academic Press, London (1988)
10. Couprie, M., Bertrand, G.: Tessellations by connection. *Pattern Recognition Letters* **23** (2002) 637–647
11. Vincent, L.: Efficient computation of various types of skeletons. In Loew, M., ed.: *Medical Imaging V: Image Processing. Volume SPIE-1445*. (1991) 297–311
12. Davies, E., Plummer, A.: Thinning algorithms: a critique and a new methodology. *Pattern Recognition* **14** (1981) 53–63
13. Pudney, C.: Distance-ordered homotopic thinning: a skeletonization algorithm for 3D digital images. *Computer Vision and Image Understanding* **72** (1998) 404–413
14. Ranwez, V., Soille, P.: Order independent homotopic thinning. *Lecture Notes in Computer Science* **1568** (1999) 337–346
15. Rosenfeld, A., Pfaltz, J.: Distance functions on digital pictures. *Pattern Recognition* **1** (1968) 33–61
16. Danielsson, P.E.: Euclidean distance mapping. *Computer Graphics and Image Processing* **14** (1980) 227–248



# A Perturbation Suppressing Segmentation Technique Based on Adaptive Diffusion

Wolfgang Middelmann, Alfons Ebert, Tobias Deißler, and Ulrich Thoennessen

FGAN-FOM Research Institute for Optronics and Pattern Recognition, Germany  
Middelmann@fom.fgan.de, Alfons.Ebert@fom.fgan.de

**Abstract.** Segmentation is a fundamental task in pattern recognition and basis for high level applications like scene reconstruction, change detection, or object classification. The performance of these tasks suffers from a distorted segmentation. In this contribution an adaptive diffusion-based segmentation method is proposed suppressing perturbations in the segmentation with focusing on small regions with high contrast to their surrounding. The algorithm determines in each step the diffusion tensor. It is re-weighted with respect to an assessment stage. A comparative study uses high-resolution remote sensing data.

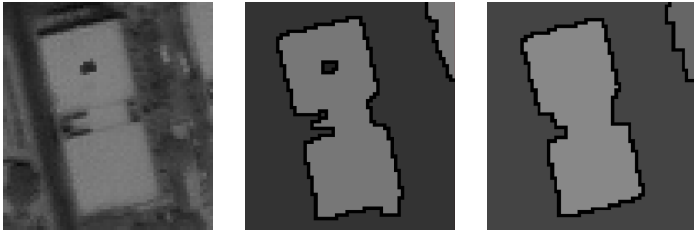
## 1 Introduction

Image Segmentation is a main task in remote sensing or medical image analysis. Common techniques [5, 10, 11] are thresholding, edge-based, and region-based methods, or use variational approaches [2] up to Markovian-based processes [4, 6, 14]. These are often combined with multi-resolution or scale-space methods [16, 17].

Image Segmentation should be the partitioning of an image into meaningful regions. However, the above mentioned techniques rely only on similarity criteria. Thus, often the user requirements are not accomplished. This induces the need for knowledge-driven segmentation. Early studies are given in [13]. Further developments for this purpose are regularization terms, model functions [1, 12], or specialized diffusion procedures [17]. For this the user has to transform its demands into suitable processing parameters or analytical terms.

This contribution deals with a typical example, the postulation of a minimal region size. One difficulty of the segmentation may be the prevention of small regions with high contrast to their surrounding. An example in Figure 1 demonstrates the unintentional influence of these areas to the resulting region structure. In our approach the suppression of small segments is an integral part of the diffusion-based segmentation.

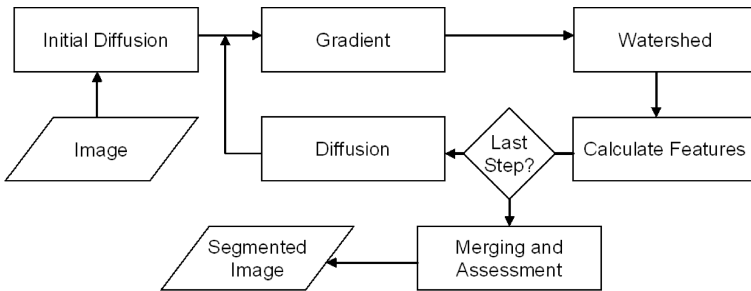
Following the approach of Vanhamel [7, 8, 16] we use a diffusion-based segmentation, see section 2. In section 3 the proposed requirement controlled segmentation is presented. A first comparative study using IKONOS satellite data is given in section 4. The conclusion resumes the contribution and gives hints for future studies.



**Fig. 1.** Original with small defect, Standard segmentation, Segmentation without defect (Original images: Courtesy of European Space Imaging / © European Space Imaging GmbH)

## 2 A Diffusion-Based Segmentation Scheme

A generic diffusion-based segmentation scheme is presented in Figure 2. The input image is diffused up to the localization scale, i.e. an edge preserving filtering is carried out. The following iteration consists of the determination of the gradient magnitude with subsequent watershed segmentation. Region features are calculated and the diffusion is accomplished. After the predefined number of iteration steps, the generated segmentations are merged and assessed with respect to the region features.



**Fig. 2.** A generic diffusion-based segmentation scheme

For comparison purposes we use the isotropic Perona-Malik diffusion (1) like Vanhamel [16]. Other edge- or coherence-enhancing schemes [17] may improve the results for particular cases as high-resolution Synthetic Aperture Radar (SAR) image analysis or medical applications.

$$\partial_t u = -\text{div} \left( g \left( |\nabla u_\sigma|^2 \right) \nabla u \right) \tag{1}$$

Hereby  $u = u(t, x, y)$  is the image at diffusion time  $t$ ,  $u_\sigma$  is the Gauss-filtered  $u$  and

$$g(s) = \begin{cases} 1 & (s \leq 0) \\ 1 - \exp \left[ \frac{-3.31488}{(s/\lambda)^4} \right] & (s > 0) \end{cases}, \quad s = |\nabla u_\sigma|^2. \tag{2}$$

We use the discrete semi-implicit scheme with additive operator splitting (AOS) technique. Further details can be found in [18].

The diffused image is segmented by applying the watershed algorithm on its gradient magnitude image. Pixels are grouped into regions and boundaries allowing the calculation of region features like region size, perimeter length, mean value, variance, or higher moments. The process stops at a pre-defined diffusion time.

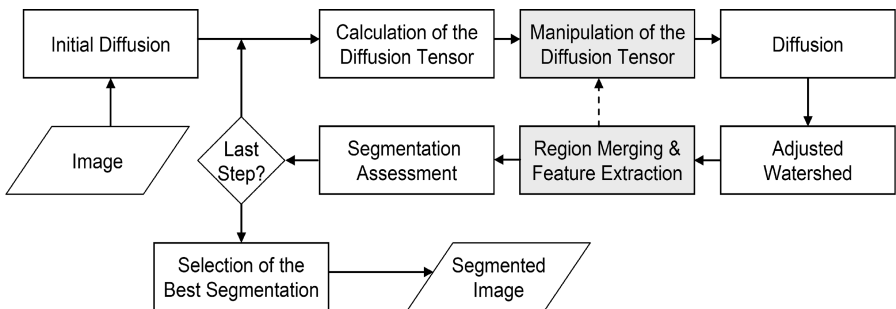
It follows the assessment-driven merging for analyzing the region features. Merging is possible within each time scale or across different scales. The global segmentation assessment concludes the processing chain.

Vanhamel [7, 8, 16] proposed to use a merging in each scale based on mean value differences, or merging based on the Dynamic of Contours (DC) [9]. The successive merging may trigger a statistical test, e.g. T-test, F-test, or  $\text{Chi}^2$ -test, defining intermediate segmentation results. To determine the best segmentation result, the so-called Liu-Yang-Value [19] is used. This heuristic assesses the global segmentation.

The discussed diffusion-based segmentation scheme does not consider any knowledge provided by the user except a few abstract input parameters.

### 3 Adaptive Segmentation

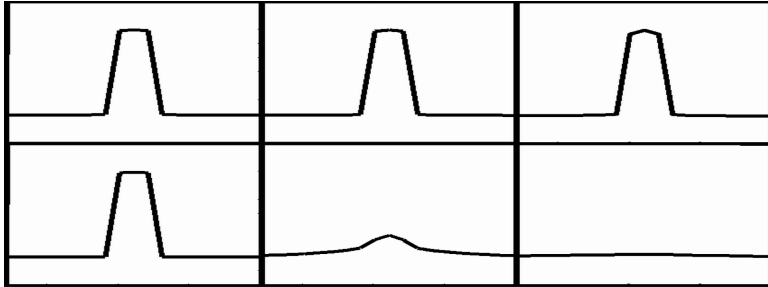
While the user postulates a minimal region size, the main difficulty is preventing small regions although these might have high contrast to their surrounding. The solution of this problem is the identification and elimination of these small segments. A direct approach could be deleting these regions by interpolation from the borders. However, this would destroy valuable information. If there are scattered objects belonging to one large region in a coarser scale these objects would be eliminated. An example of a backscatter row in a PAMIR SAR image [3] is depicted in Figure 9. Therefore, a solution is proposed that modifies the diffusion process by manipulating the diffusion tensor. A flowchart of the processing chain can be seen in Figure 3.



**Fig. 3.** Flowchart of the proposed segmentation process

In comparison to the general processing scheme presented in section 2 the requirement controlled segmentation is enabled by modifications that take place in the “Adjusted Watershed” and the “Manipulation of the Diffusion Tensor” modules. In contrast to the user defined minimal region size, the diffusion-based segmentation preserves small regions with high contrast. For example, if streets and buildings are of interest, then cars on the street should be ignored. Our proposed solution is to

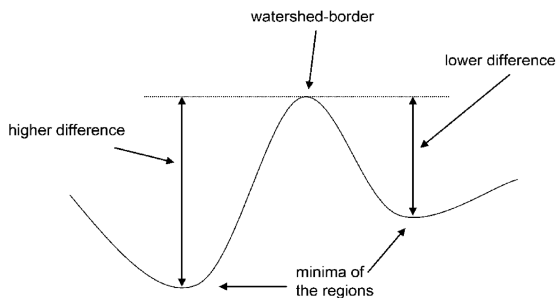
re-weight the diffusion tensor. At the borders of such small regions the diffusion tensor is set to the maximum value  $1$  of  $g$  (eq. (2)). Thus, the diffusion damps their contrast. Large regions are unaffected. Figure 4 shows the different behavior of the standard Perona-Malik diffusion and the enhanced diffusion scheme. While diffusing small regions it is possible that some of them will merge and form a bigger region. Thus, scattered structures are forced to emerge.



**Fig. 4.** Upper: Perona-Malik diffusion, Lower: enhanced diffusion (at  $\tau=1, 4, 9, \lambda=0.01$ )

At low diffusion times  $t$  the watershed segmentation often yields in an over-segmentation. In spite of this behavior our approach identifies the small segments to be eliminated. It is supported additionally by the h-minima transform [15] used to suppress shallow local minima. Especially, the h-minima transform is appropriate to denoise areas of quasi constant gradient.

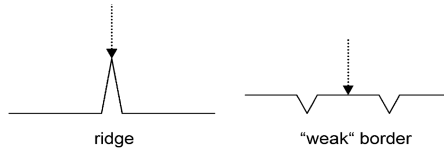
The final identification of the small region is obtained by region merging. For the region merging, two processes are implemented: Merging based on mean value differences, and merging based on the Dynamic of Contours (DC) [9]. The former is performed by calculating the distance between the mean gray values of adjacent regions or the Euclidian distance of the mean RGB-values, if color images are processed. Merging starts with the smallest distance and is continued until a threshold is reached. The DC measures the strength of the ridge separating two regions. It is determined as minimum over all related contour point dynamic values. In Figure 5 the dynamic of an arbitrary contour point is explained.



**Fig. 5.** Schematic view of the Dynamic of a Contour Point

This dynamic is calculated as the minimum of the height differences between the border point and the minima of the two adjacent regions. Therefore, the DC is the dynamic of a saddle point on the watershed border. The greater this value, the less is the probability that these regions belong together. The merging process starts by deleting the border with the smallest DC value and recalculates the DC values of the adjacent regions. This step is repeated until a pre-defined threshold is reached. Small regions with high contrast are very robust against both merging schemes. Thus, the desired regions can be identified.

The watershed segmentation results in borders at ridges of the gradient magnitude and in so-called weak borders consisting of pixels on unincisive ridges, these structural phenomena are depicted in Figure 6.

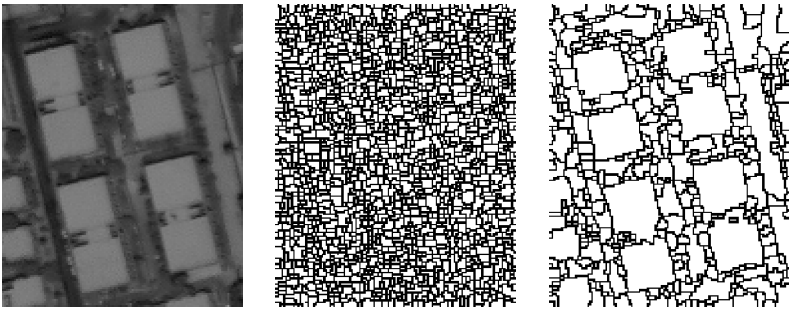


**Fig. 6.** Different watershed borders in the gradient magnitude

In addition to the similarity or DC intended merging, a merging of segments with such weak watershed borders is implemented. A further problem is that real segment borders may consist of ridge-pixels and weak pixels. Therefore, our method decides about a possible merging by assessing the size of connected components of these weak pixels related to user requirements. Merging these regions is a fast and reliable way of improving the segmentation result and is a good addition to the h-minima transform.

## 4 Experimental Results

Our diffusion segmentation method was tested using IKONOS satellite data. Figure 7 depicts an aerial image of a built-up area. The watershed segmentation applied on this original image results in oversegmentation. The h-minima transform [15] is used to suppress shallow local minima to denoise areas of quasi constant gradient.



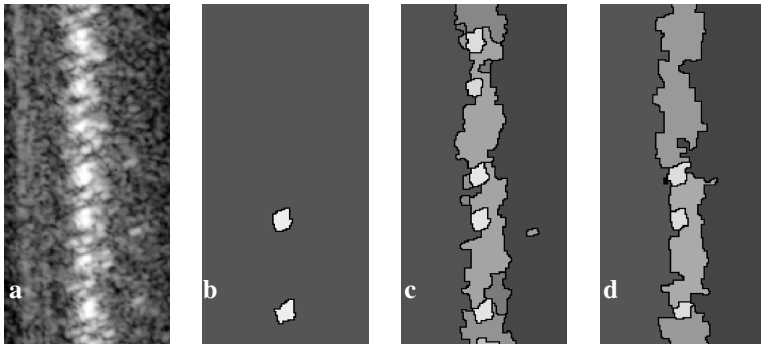
**Fig. 7.** Left: original IKONOS image of buildings, Middle: watershed results in oversegmentation, Right: watershed after denoising with h-minima transform (Original images: Courtesy of European Space Imaging / © European Space Imaging GmbH)

Our segmentation scheme is supported by the h-minima transform. Using only the merging module comparable results are obtained. However, the calculation time of the whole process with h-minima transform is reduced.



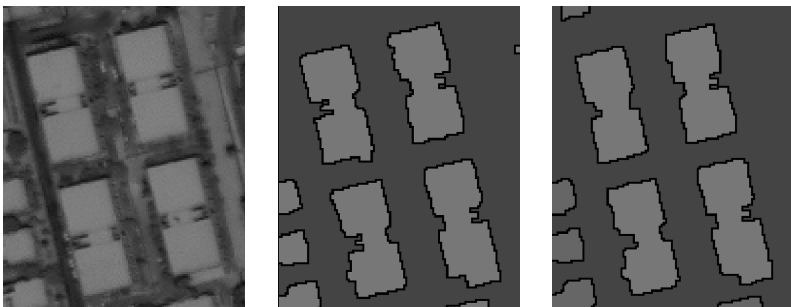
**Fig. 8.** Left: Test image, Middle: Vanhamel approach, Right: Our segmentation result

Figure 8 presents an example of scattered objects. The Vanhamel approach leaves small objects unchanged while our approach combines them to a large region. The results for the above introduced +SAR example are given in Figure 9. The Vanhamel approach with LYV-based segmentation selection yields unsatisfactory results. The manually selected one is comparable to our result, but has a few more small segments.



**Fig. 9.** **a:** Original SAR image © FGAN-FHR, **b:** Vanhamel approach result at LYV optimum, **c:** Vanhamel approach – result manually selected, **d:** Our approach at LYV optimum

The two different approaches give comparable results for the IKONOS image with four main buildings, depicted in Figure 10.



**Fig. 10.** Left: Original IKONOS image data, Middle: Vanhamel approach result at LYV optimum, Right: Our approach at LYV optimum

Taking into account the user requirement concerning the minimal region area, it is possible to suppress small details as shown in Figure 11. This behavior supports the automatic structural scene analysis, especially the reconstruction of built up areas.



**Fig. 11.** Left: Industrial building in IKONOS data, Middle: Vanhamel approach result at LYV optimum, Right: Our result at LYV optimum

## 5 Conclusion

The manipulation of the diffusion tensor has proven to be a good method to control the segmentation process. In this contribution the minimal region size has been chosen as an example for integrating user requirements into a segmentation scheme. It is possible to suppress small segments in spite of a high contrast. Scattered objects may be combined to larger ones. The h-minima transform decreases the computational effort. The segmentation consistency is improved by erasing so-called weak watershed borders that may cause artifacts. Two merging strategies are tested, the mean-value-based and the DC-based, giving comparable results. The best method depends on the sensor data type. The separation of diffusion and identification of undesirable segments boosts the approaches efficiency.

The approach proposed by Vanhamel is used for comparisons. It differs in the merging strategy and in the handling of small undesired regions. The results of our tests often give similar results, but in special cases as scattered objects the new method has some advantages, especially for the analysis of SAR images. The LYV heuristic for selecting the best segmentation result fails for several examples. The new approach suppressing small regions seems to be more consistent with the LYV assessment. However, further test are necessary for proving this behavior. An improvement of the LYV heuristic is desirable.

Future activities may be the development of further methods for the manipulation of the diffusion tensor. The user requirements specification should be extended up to complex generic models for the merging strategy. Thus, the segmentation can be focused on man-made-objects.

**Acknowledgments.** The authors wish to thank I. Vanhamel at Vrije Universiteit Brussel in Belgium for providing the software package.

## References

1. T. Brox, B. Rosenhahn, J. Weickert, Three-Dimensional Shape Knowledge for Joint Image Segmentation and Pose Estimation, Proceedings 27th DAGM Symposium, Wien, Austria, September 2005, W. G. Kropatsch, R. Sablatnig, A. Hanbury (eds.), LNCS 3663, (2005) pp. 109-116, Springer
2. D. Cremers, C. Schnörr and J. Weickert, Diffusion-Snakes Combining Statistical Shape Knowledge and Image Information in a Variational Framework, N. Paragios (ed.), IEEE Intl. Workshop on Variational and Levelset Methods, Vancouver, (2001) pp. 137-144.
3. J. H. G. Ender, A. R. Brenner, "PAMIR—a Wideband Phased Array SAR/MTI System," IEE Proc. Radar Sonar Navigat. 150 (3), (2003) 165–172.
4. F. R. Hansen, H. Elliott, Image Segmentation Using Simple Markov Random Field Models, Computer Graphic and Image Processing, vol. 20, (1982) pp 101-132.
5. R. M. Haralick, L. G. Shapiro, Survey- image segmentation techniques, Computer Vision Graphics and Image Processing, vol. 29, (1985) pp. 100-132.
6. I. Kovtun, Partial optimal labeling search for a NP-hard subclass of (max,+) problems, Proceedings 25th DAGM Symposium, Magdeburg, Germany, September 2003, G. Krell, B. Michaelis (eds.), LNCS 2781, (2003) pp 402-409, Springer.
7. S. Makrogiannis, I. Vanhamel, S. Fotopoulos, H. Sahli, Scale Space Segmentation of Color Images Using Watersheds and Region Fusion, IICIP 2001, Thessaloniki-Greece, (2001)
8. S. Makrogiannis, I. Vanhamel, H. Sahli, Scale space Segmentation of Color Images, TR-0076, Vrije Universteit Brussel, (2001)
9. L. Najman, M. Schmitt, Geodesic Saliency of Watershed Contours and Hierarchical Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 12, (1996) pp. 1163-1173.
10. H. Niemann, Pattern Analysis and Understanding, Springer, (1990)
11. N. R. Pal, S. K. Pal, A review on image segmentation techniques, Pattern Recognition, vol. 26, no. 9, (1993) pp. 1277-1294
12. M. Rousson, N. Paragios, Shape Priors for Level Set Representation, In A. Heyden, G. Sparr, M. Nielsen, P. Johansen (eds.), Computer Vision – ECCV 2002, LNCS 2351, (2002) pp. 78-92, Springer
13. G. Sagerer, H. Niemann, Semantic Networks for Understanding Scenes, Plenum Press, 1997
14. M. I. Schlesinger, V. Hlavác, Ten Lectures on Statistical and Structural Pattern Recognition, Kluwer Academic Publishers, Dordrecht, 2002
15. P. Soille, Morphological Image Analysis: Principles and Applications, Springer, 2003
16. I. Vanhamel, H. Sahli, I.Pratikakis, Hierarchical Multiscale Watershed Segmentation of Color Images, Proceedings of First International Conference on Color in Graphics and Image Processing, Saint-Etienne - France, (2000) pp. 93-100
17. J. Weickert, Anisotropic Diffusion in Image Processing, B. G. Teubner, (1998)
18. J. Weickert, Efficient and Reliable Schemes for Nonlinear Diffusion Filtering, IEEE Transaction on Image Processing, Vol. 7, No. 3, (1998)
19. Y.-H. Yang, J. Liu, Multiresolution Image Segmentation, IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 16, (1994) pp. 689-700



# Weighted Order Statistic Filters for Pattern Detection

Slawomir Skoneczny and Dominik Cieslik

Institute of Control and Industrial Electronics, Warsaw University of Technology,  
ul. Koszykowa 75, 00-662 Warszawa, Poland  
slaweks@isep.pw.edu.pl, d.cieslik@iem.pw.edu.pl

**Abstract.** In this paper we propose a method of using Weighted Order Statistic (WOS) filters for the task of pattern detection. Usually WOS filters are applied to noise removal. An efficient algorithm for pattern detection is described in details with emphasis put on the problem of a proper choice of filter windows. Also practical results of different pattern detection cases are presented.

## 1 Introduction

Detection of characteristic patterns (of predefined shape) in a single image or in an image sequence processing is a very important task. There are several methods of solving this problem but many of them require pattern classification methods with feature vectors. One of a few "non-classifying" possibilities is an approach based on mathematical morphology with markers and geodesic operations, but this is rather time consuming and sometimes requires assistance of human operator. In this paper we propose a novel method of detection of characteristic patterns taking advantage of the theory of Weighted Order Statistic filters and their properties. Usually this kind of filters is applied to image removal, however some papers devoted to the usage of WOS filters in pattern recognition area can be found, e.g. [8].

## 2 Application of the WOS Filter to Detection of Patterns in Image

### 2.1 Some Definitions

Let  $\mathbf{A}$  denotes an input image. The filter window will be denoted as  $\Omega_{n,m}^{(a,b)}$ , where  $(a, b)$  are coordinates of the window center according to the beginning of the image,  $n \times m$  – size of the window. For a pixel  $(i, j)$  of the image the filtering window is defined as:

$$\Omega_{n,m}^{(a,b)} = \{(i+l, j+k) \in \mathbf{A} : -a < l \leq n-a \quad \text{and} \quad -b < k \leq m-b\} . \quad (1)$$

**Definition 1.** The stack filter on the 2-D signal  $\mathbf{I}$  is defined as [2]:

$$S(\mathbf{I}) = \sum_{n=1}^N f(\mathbf{B}^n),$$

where  $\mathbf{I}_{i,j} \in \{0, 1, \dots, N\}$ ,  $\mathbf{B}_{i,j}^n \in \{0, 1\}$ . This class of filters possesses so called "stacking property", which guarantees that the binary signals  $\mathbf{B}^n$  filtered with the function  $f$  stack.

**Definition 2.** A boolean function is called the Positive Boolean Function (PBF) if it does not have any complements of variables. For each  $f$  being a PBF with binary matrices  $\mathbf{a}$  and  $\mathbf{b}$  as its arguments it can be written that:

$$\mathbf{a} \geq \mathbf{b} \Rightarrow f(\mathbf{a}) \geq f(\mathbf{b}),$$

where  $a_{i,j}, b_{i,j} \in \{0, 1\}$  and  $f \in \{0, 1\}$ .

**Definition 3.** In the integer domain the output of a WOS filter is calculated by duplicating each input sample  $X_i$  to the number of the corresponding weight  $W_i$ , sorting the resulting array of  $\sum W_i$  points and then choosing the  $T$ -th largest value from the sorted vector. Here the weights  $W_i$  and the threshold  $T$  are restricted to be positive integers. Usually the WOS filter is denoted as  $\langle W_1, W_2, \dots, W_N; T \rangle$ . A procedure of finding the PBF corresponding to WOS filter is described in [1]. A very important class of WOS filters are Weighted Median Filters [3].

## 2.2 Pattern Detection and Their Removal

The main task of WOS filters is denoising. We show that this class of tools can also be used for detection of patterns of some specific shape. However, this task is usually more complicated than the standard process of noise removal. The difficulty is in the fact that the WOS filter preserves by definition only those gray levels matching the certain pattern (which have been chosen earlier). For the case presented in Fig. 1 both the pattern itself as well as all the objects containing this pattern will be detected. Usually in such cases an image processing engineer expects preserving all the objects matching precisely this pattern or those ones with an assumed amount of deviation from the perfect pattern. Therefore the results obtained by using the standard procedures of object removal are unsatisfactory. The idea of detecting objects of a given shape is as follows. Let us assume that the filter window is denoted by  $\Omega$ . Two sets  $\mathbf{a}$  and  $\mathbf{b}$  are constructed such as:

$$x_{i,j} \in \Omega, \mathbf{a} = \{(i, j) : x_{ij} = 1\}, \mathbf{b} = \{(i, j) : x_{ij} = 1\}, \mathbf{a} \cap \mathbf{b} = \emptyset, \mathbf{b} \subset \mathbf{a}',$$

where  $\mathbf{a}'$  means a complement of  $\mathbf{a}$ . The whole window defines the argument matrix for the function  $f(\Omega)$  ( $f$  is a PBF), for which the WOS filter will be designed. The set  $\mathbf{a}$  defines a pattern that will be detected for a single gray level, and the set  $\mathbf{b}$  defines those windows elements, that are undesirable in the detected pattern.

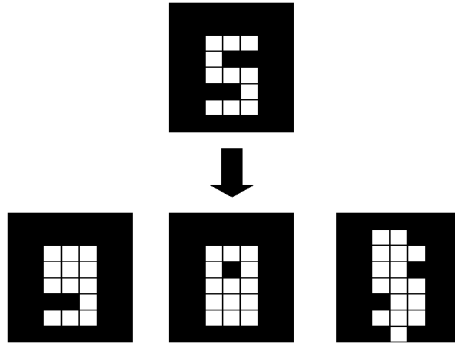


Fig. 1. A perfect and detected patterns

### 2.3 The Algorithm for Pattern Detection

The algorithm for pattern detection [4] is based on those two sets **a** and **b** which define filtering windows **a** and **b** and on the additional filter used for noise removal. Such a filter is given by the PBF  $f(\mathbf{x}) = x_0x_2 + x_1x_2 + x_3x_2 + x_4x_2$  (WOS filter  $\langle 1, 1, 4, 1, 1; 5 \rangle$ ) and by the window **c** presented in Fig. 2. After application of this filter the isolated pixels lighter than their neighborhood will be removed. The detailed algorithm for detection of patterns that are described



Fig. 2. A filter window **c** (central pixel (2,2))

by two windows **a** and **b** is as follows:

STEP 1: Detection of elements of the size equal to the size of the pattern or greater

- a)  $S_a(\mathbf{A}) = \mathbf{B}$ ,
- b)  $S_c(\mathbf{B}) = \mathbf{D}$ ,
- c)  $\mathbf{B} - \mathbf{D} = \mathbf{E}$

STEP 2: Detection of elements that are of the size equal to the pattern or smaller

- a)  $\text{inv}(\mathbf{A}) = \mathbf{F}$ ,
- b)  $S_b(\mathbf{F}) = \mathbf{G}$ ,
- c)  $S_c(\mathbf{G}) = \mathbf{H}$ ,
- d)  $\mathbf{G} - \mathbf{H} = \mathbf{I}$

STEP 3: Choosing the elements that match the pattern

- a)  $\text{inf}(\mathbf{I}, \mathbf{E}) = \mathbf{J}$ ,

where:

- A**– the input frame,
- $S_a(\mathbf{A})$ – the result obtained by applying the WOS filter of window **a** to frame **A**,
- $\text{inv}(\mathbf{B})$ – the negative of a frame **B**,
- $\text{inf}(\mathbf{I}, \mathbf{E})$  – the operation of infimum of two images **I** and **E**.

This algorithm consists of three stages: detection of objects of a shape of the pattern or greater, detection of the objects of a shape of the pattern or smaller and at last finding the intersection of these two sets (i.e. finding the elements that precisely match the perfect pattern). At the first stage the frame **A** is processed by the WOS filter of the window **a**. As a result of this processing the objects that are smaller than the pattern are removed. Next, the frame **B** is processed by the WOS filter window **c**. As a result of this action pixels with no neighborhood (so these ones that are interesting to us) are left in image. In order to detect them frame **B** is subtracted from the frame **D**. This way we obtain the pixels that are in detected patterns or in objects including those patterns. At the second stage the elements equal to size of the pattern or smaller ones should be preserved. In order to achieve this goal it is necessary to find the frame **F**– a negative of the input frame **A**. After having processed this negative frame by the WOS filter of the window **b** the elements smaller or exactly equal to the pattern are removed. In order to detect them the frame **H** is subtracted from the frame **G** and this way we have the frame **I**. At the third stage pixels corresponding to the searched shape are found.

### 2.4 The Choice of Windows **a** and **b**

The filter window **a** is responsible for the "minimum of pattern" that will be searched. The filter window **a** defines its maximal size. If the window **b** is exactly a complement of the window **a** then objects that perfectly match the pattern will be detected. However if the window **b** will be smaller than the complement of the window **a** then a group of shapes "between" **a** and **b'** will be found. Some possibilities are presented below. It is clear that the higher the difference between both windows the more different shapes will be detected. Here **b'** means the complement of **b**.

## 3 Practical Example

### 3.1 Example

Here is an example showing the effects of the algorithm for the window **a** (PBF  $f(\mathbf{x}) = x_0x_1x_2x_3x_4x_5x_6x_7x_8$ , WOS  $\langle 1, 1, 1, 1, 1, 1, 1, 1, 1, 9 \rangle$ ) presented in Fig. 7 and the window **b** (PBF  $f(\mathbf{x}) = \prod_{i=0}^{15} x_i$ , WOS  $\langle 1 \diamond 15; 15 \rangle$ ), which is its complement.

## 4 Results of the Experiments

### Experiment 1

The cross-shaped object has been detected perfectly in the frame **I** because its brightness is significantly higher than the brightness of other detected pixels

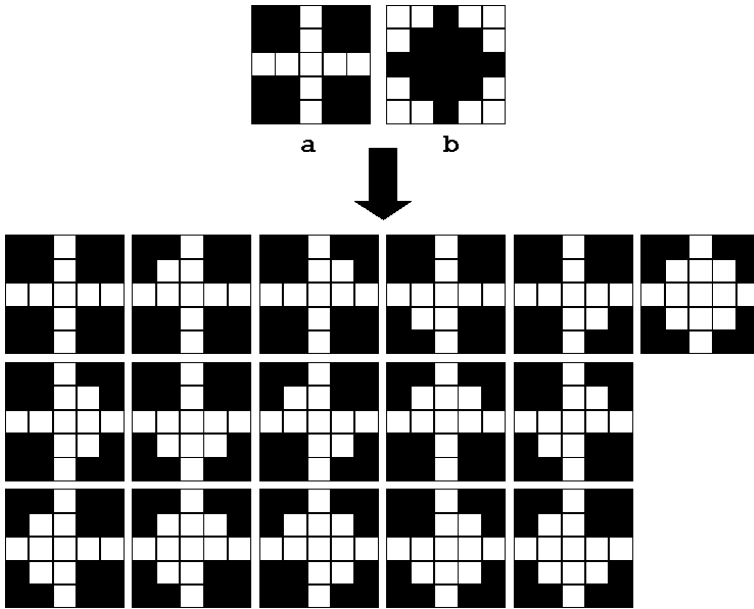


Fig. 3. Windows a and b with possible detected patterns

Table 1. PBF's and the WOS filters corresponding to them for Experiment 2 (◊ means repetition operator)

Filter I		
Type of window	PBF	WOS
$\Omega_{aI}^{(4,4)}$	$f(\mathbf{x}) = \prod_{i=0}^{12} x_i$	$\langle 1 \diamond 12; 12 \rangle$
$\Omega_{bI}^{(4,4)}$	$f(\mathbf{x}) = \prod_{i=0}^{35} x_i$	$\langle 1 \diamond 35; 35 \rangle$
$\Omega_{cI}^{(2,2)}$	$f(\mathbf{x}) = x_0x_2 + x_1x_2 + x_3x_2 + x_4x_2$	$\langle 1, 1, 4, 1, 1; 5 \rangle$
Filter II		
Type of window	PBF	WOS
$\Omega_{aII}^{(4,4)}$	$f(\mathbf{x}) = \prod_{i=0}^{12} x_i$	$\langle 1 \diamond 12; 12 \rangle$
$\Omega_{bII}^{(4,4)}$	$f(\mathbf{x}) = \prod_{i=0}^{23} x_i$	$\langle 1 \diamond 23; 23 \rangle$
$\Omega_{cII}^{(2,2)}$	$f(\mathbf{x}) = x_0x_2 + x_1x_2 + x_3x_2 + x_4x_2$	$\langle 1, 1, 4, 1, 1; 5 \rangle$
Filtr III		
Type of window	PBF	WOS
$\Omega_{aIII}^{(5,5)}$	$f(\mathbf{x}) = \prod_{i=0}^{17} x_i$	$\langle 1 \diamond 17; 17 \rangle$
$\Omega_{bIII}^{(5,5)}$	$f(\mathbf{x}) = \prod_{i=0}^{10} x_i$	$\langle 1 \diamond 10; 10 \rangle$
$\Omega_{cIII}^{(2,2)}$	$f(\mathbf{x}) = x_0x_2 + x_1x_2 + x_3x_2 + x_4x_2$	$\langle 1, 1, 4, 1, 1; 5 \rangle$

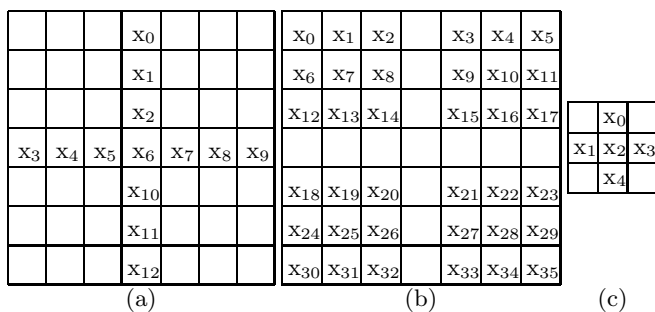


Fig. 4. Windows of the filter I (a)  $\Omega_{aI}^{(4,4)}$  (b)  $\Omega_{bI}^{(4,4)}$  (c)  $\Omega_{cI}^{(2,2)}$  – Experiment 2

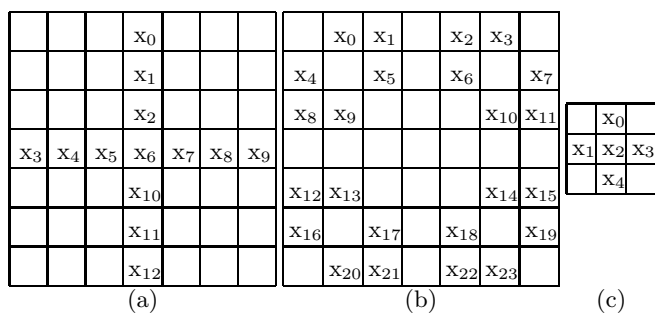


Fig. 5. Windows of the filter II (a)  $\Omega_{aII}^{(4,4)}$  (b)  $\Omega_{bII}^{(4,4)}$  (c)  $\Omega_{cII}^{(2,2)}$  – Experiment 2

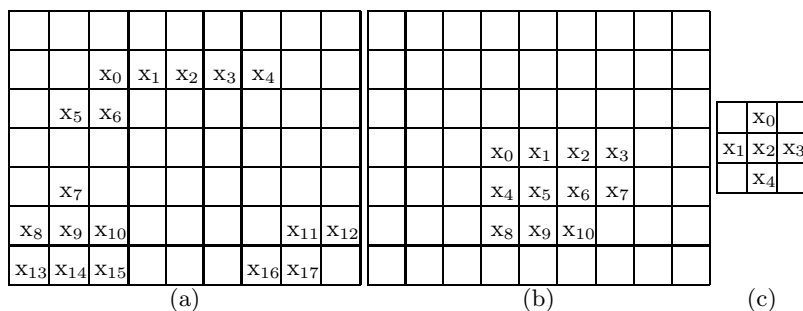


Fig. 6. Windows of the filter III (a)  $\Omega_{aIII}^{(5,5)}$  (b)  $\Omega_{bIII}^{(5,5)}$  (c)  $\Omega_{cIII}^{(2,2)}$  – Experiment 2

shape of which was similar to the searched pattern. For the frame **J** the result is also very good. Three objects have been detected of the shapes between the cross and the asterisk. The other objects of a similar shape are also detected but the visible effect is not so clear. The objects that partially match the searched object (in this meaning that the reference object is a perfect pattern) are also detected.

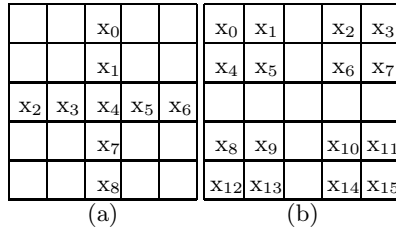


Fig. 7. (a) Window a (b) Window b – Experiment 1

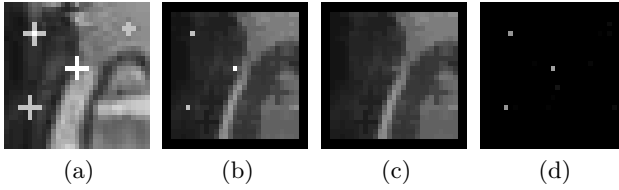


Fig. 8. Frames (a) A, (b) B, (c) D, (d) E – Experiment 1

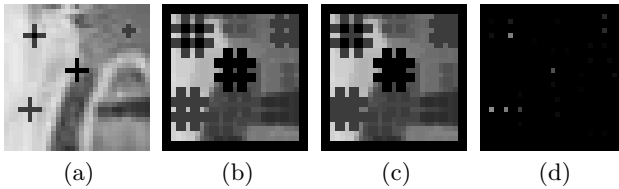


Fig. 9. Frames (a) F, (b) G, (c) H, (d) I – Experiment 1

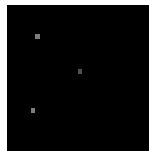


Fig. 10. A Frame J – Experiment 1

If the considered object matches partially the detected one, then the object is detected to such extend to which its highest level of gray will completely matches the perfect pattern. So the results can be interpreted in the following way: the brighter is the point in the output frame, the more similar to the perfect pattern.

Experiment 2

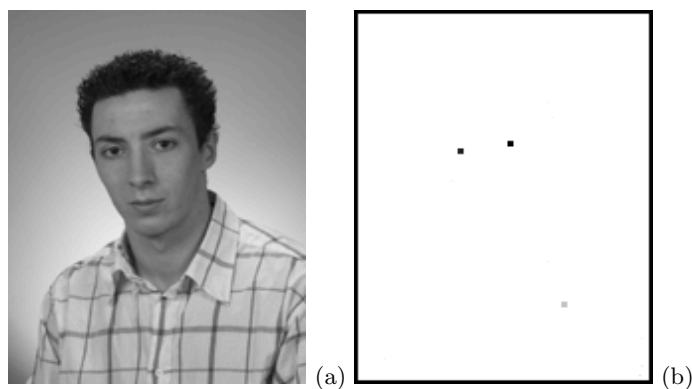
Predefined objects are detected in the frames: I and J. These frames are presented in respective figures. The windows  $\Omega_a$ ,  $\Omega_b$  and  $\Omega_c$  as well as the PBF's



**Fig. 11.** (a) Frame **I** (b)  $S_{\Omega_{aI}^{(4,4)}, \Omega_{bI}^{(4,4)}, \Omega_{cI}^{(2,2)}}(\mathbf{I})$  – Experiment 2



**Fig. 12.** a) Frame **J** b)  $S_{\Omega_{aII}^{(4,4)}, \Omega_{bII}^{(4,4)}, \Omega_{cII}^{(2,2)}}(\mathbf{J})$  – Experiment2



**Fig. 13.** (a) Frame **K** (b)  $S_{\Omega_{aIII}^{(5,5)}, \Omega_{bIII}^{(5,5)}, \Omega_{cIII}^{(2,2)}}(\mathbf{K})$  – Experiment 2

corresponding to them and also the WOS filters are presented further. The filter I detects the objects that exactly match the little cross-pattern. The filter II detects those elements of shape that is between the shape of little cross-pattern and the little asterix-pattern. The filter III with complicated windows detects both eyes of the person in Fig. [13](#).



## 5 Summary and Conclusions

WOS filters have been considered mainly as filtering tool for 15 years by now [5]. At the present time even the specialized hardware for WOS filters in sophisticated forms including neural network architecture is available [7]. In addition to that an extension of the concept of the WOS filters called Permutation Weighted Order Statistic filters (which are in essence time-varying WOS filters) has been proposed [6]. In our paper we have demonstrated that the WOS filters widely used in image processing for denoising purpose can also be used quite effectively for detection of characteristic pattern in image. Due to the threshold decomposition and the stacking property of WOS filters this application is efficient and the results are in most cases satisfying. The only inconvenient feature of this type of approaches is a necessity of defining large windows for big objects that have to be detected.

## References

1. Yli-Harja, O., Astola, J., Neuvo, Y.: Analysis of the Properties of Median and Weighted Median Filters Using Threshold Logic and Stack Filter Representation. *IEEE Trans. Signal Processing*, vol. 39, no. 2, 1991, 395–417
2. Yu, P. T., Liao, W. H.: Weighted Order Statistics Filters - Their Classifications, Some Problems and Conversion Algorithm. *IEEE Trans. Signal Processing*, vol. 42, no. 10, 1994, 2678–2691
3. Yin, L., Yang, R., Gabbouj, M. Neuvo, Y.: Weighted Median Filters: A Tutorial. *IEEE Trans. Circuit and Systems -II: Analog and Digital Signal Processing*, vol. 43, no. 3, March 1996, 157–192
4. Cieslik, D.: Nonlinear Filters in image processing. M. Sc. Thesis, Warsaw University of Technology, 2003 (in polish)
5. Marshall, S.: New Direct Design Method for Weighted Order Statistic Filters. *IEE Proc.-Vis. Image Signal Process.*, vol. 151, no. 1, February 2004, 1–8
6. Arce, G., Hall, T., Barner, K.: Permutation Weighted Order Filter Lattices. *IEEE Trans. Image Processing*, vol. 4, no. 8, 1995, 1070–1083
7. Kowalski, J.: Weighted Order Statistic Filter Chip Based on Cellular Neural Network Architecture. *Proc. of IEEE Int. Conf. Image Processing (ICIP03) 2003*, vol II, 575–578
8. Porter, R., Eads, D., Hush, D., Theiler, J.: Weighted Order Statistic Classifiers with Large Rank-Order Margin. *Proc. of International Conference in Machine Learning*, August 2003

# Real-Time Image Segmentation for Visual Servoing\*

Witold Czajewski<sup>1</sup> and Maciej Staniak<sup>2</sup>

<sup>1</sup> Institute of Control and Industrial Electronics, Warsaw University of Technology,  
ul. Koszykowa 75, 00-662 Warszawa, Poland  
W.Czajewski@isep.pw.edu.pl

<sup>2</sup> Institute of Control and Computation Engineering, Warsaw University of Technology,  
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland  
mstaniak@elka.pw.edu.pl

**Abstract.** Precise and real-time segmentation of color images is a crucial aspect of many applications, where high accuracy and quick response of a system is required. There is a number of algorithms available, but they are either fast or accurate, but not both. This paper describes a modification to one of the fastest color segmentation methods based on constant thresholds. Our idea is to use variable thresholds and merge the results achieving higher precision of color segmentation, comparable with adaptive methods. Using variable thresholds requires multiple passes of the algorithm which is time consuming, but thanks to our modification the processing time can be reduced by up to 50%. The original algorithm as well as the proposed modification are described. The performance of the modified method is tested in a real-time visual servoing application.

## 1 Introduction

Region segmentation of color images is a crucial step in many vision applications. During that process individual pixels of an image are classified into one of a finite number of color classes and pixels belonging to the same class are grouped together for further high level processing. The most common solutions to this problem are: linear color thresholding, nearest neighbor classification, color space thresholding and probabilistic methods [2]. These approaches enable either precise (or human-like) color segmentation at low speed or real-time<sup>1</sup> processing with relatively poor accuracy (in comparison with human performance). In certain cases, however, both accuracy and speed must be provided for.

An interesting answer to this problem was proposed in [2], which was an inspiration of our work. We decided to use this method for feature extraction in a visual servoing system. However, since the original method does not allow for color variations and defines colors as constant rectangular blocks in the color space, it often fails to properly separate neighboring regions of the same color. In our work, we modify the original method so that it is far more robust to subtle color changes. Our approach is equivalent to applying the original method several times, but without the

---

\* This paper was funded by MNIi grant no. 4 T11A 003 25.

<sup>1</sup> By “real-time” we understand processing of a full frame (768x576) at 25 Hz.

most computationally expensive part of the algorithm (color classification), which yields a significant performance gain. In the next section the outline of the original method is presented. The following sections describe our modification and its performance in a visual servoing system.

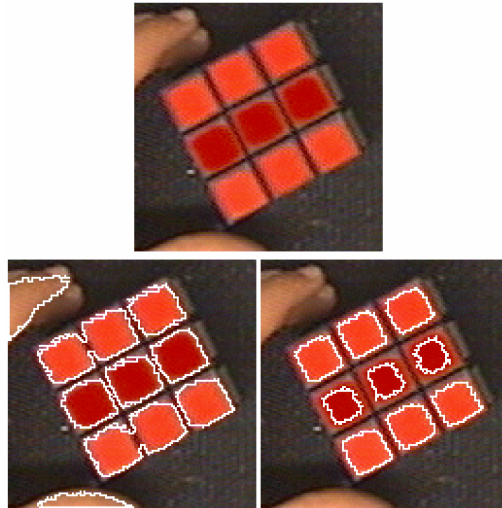
## 2 The Original Algorithm

Several color spaces are in use in color vision applications, however the most popular are such three dimensional spaces where chrominance is coded in two of the dimensions and intensity is coded in the third (HSV, HSI, YUV etc.). These transformations minimize the most significant correlations between colors and the resulting space can be partitioned into separate hyperrectangles for all the color classes by constant value thresholding in each dimension. Thus a color class is defined by a set of six thresholds: a bottom and a top value for each dimension. In order to classify a pixel, six comparisons are necessary (in the most naive approach) for each color class. In case of many color classes the number of comparisons grows rapidly. Obviously, it can be reduced by applying some decision tree optimization, but still it requires multiple comparisons per pixel which is time consuming, especially for high resolution images.

The implementation proposed in [2] uses a boolean valued decomposition of the multidimensional threshold, which is stored in arrays (usually only three dimensions are used). There are as many array elements as there are values of each color components (usually 256). Therefore class membership of a pixel can be computed as the bitwise AND operation of the elements of each array (that makes two AND operations for a three dimensional color space). In the described implementation each array element is a 32-bit integer. Hence it is possible to evaluate membership to 32 color classes with just two AND operations. This is an enormous performance gain in comparison to classical approaches where several conditions must be checked for each pixel. After the color classification and before region merging with an efficient tree-based union-find with path compression method the classified image is run length encoded. This operation can speed connected components analysis in many robotic applications where images contain relatively large areas of pixels in the same class (large objects of the same color). However in case of noisy images or many little objects the RLE compression might have the opposite effect.

The undisputable advantage of the described algorithm is its fast performance and linear scalability with the number of pixels and color space dimensions. The drawback of the method lies in the fact that the thresholds defining color classes are constant and the method is unable to cope with certain dynamic images. Consider the following example illustrating the problem. Suppose we have an image with a number of adjacent but separate red objects. Some of the objects are dark, some are bright. That should not be a problem for a color space like HSV or YUV, but the actual results may differ from expected. If we define a color class to span over both bright and dark reds, dark red objects will be properly segmented, but some bright red objects might be segmented into one big area. This is mainly due to color bleeding effect that causes pixels separating bright areas to acquire their color to some extent. If we decide to divide the color class into two subclasses, one for dark reds and the

other for bright reds, the segmentation result could be correct. However, there might always be red objects of varying intensity such that some of their pixels belong to one class and some to the other. In the end such objects will not be correctly segmented. See Fig. 1 for illustration. It is possible to select correct thresholds for one image or even for a number of images, provided the lighting conditions and observed objects do not change too much. However, in most cases, especially in dynamic robotic applications, these conditions cannot be met and there will always be some regions that are segmented incorrectly. This can lead to misinterpretation of the observed scene and can severely influence the robustness of a system.



**Fig. 1.** Original image (*top*) and two incorrect segmentation results obtained with one color class (*left*) and two color classes (*right*) for the red color. Boundaries of segmented regions are marked white, small regions are not depicted as they are treated as noise. Notice merging of segments in the left image caused by color bleeding. Although all the segments in the right image are disjoint, the dark regions are not detected correctly as the bottom threshold for dark reds was set too high.

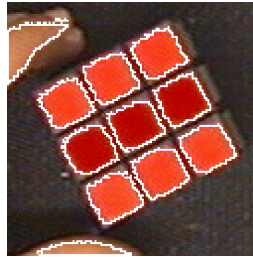
### 3 Our Approach

Our first solution to the problem was based on adaptive thresholding. The thresholds defining color classes are dynamically adjusted accordingly to local line histograms calculated in a region of interest (ROI) after initial (coarse) segmentation. This allows us to modify the thresholds based on peaks and valleys in a number of line histograms traversing the ROI. Although this method is quite successful, it is far too slow for real-time applications requiring full resolution image analysis.

Another solution which we tested involves multiple classes for a single color. Since some of the classes are overlapping (to avoid problems on class boundary mentioned in the previous section) the segmentation routine must be run at least two times (overlapping classes had to be processed separately to avoid ambiguity). This

time the processing speed is close to real-time (for two passes) and the segmentation results are satisfactory, but still not as good as we expected.

We found that using three to five subclasses for a single color gives good results, but sometimes leads to partitioning of regions with color gradients. Such a situation can occur quite often in real-life situations (colors are never solid) especially around edges. This affects mainly small objects, where the interior would belong to one class and the exterior to another, while human segmentation clearly indicates a single color class. This gave us an idea of gradual “inflating” of color classes for one color. Instead of segmenting all the color classes at once and merging the brighter and the darker segments together (which is not so obvious) we apply the algorithm a few times, each time lowering the bottom intensity threshold. In this way we start with only the brightest regions segmented and “inflate” them with each step of the routine (as the darker surroundings are gradually added each time the bottom threshold is changed). We end up with an excess number of segments (usually a few segments per region) that must be filtered out. A simple filter leaving only the second largest segment within a given radius is good enough. The segmenting results are very good (see Fig.2), however the execution time grows substantially – the entire algorithm must be repeated multiple times over the entire image.



**Fig. 2.** Almost human-like segmentation results using our modified approach and five thresholds. Boundaries of segmented regions are marked white, small regions are not depicted as they are treated as noise.

Looking carefully into the original algorithm we found that its first and usually the most time consuming part, namely color classification of all the pixels in the image can be performed only once with minimal changes to the rest of the segmentation routine. This time all the subclasses (thresholds) are considered simultaneously. The first pass of the algorithm is identical as in the original method, whereas consecutive passes omit the color classification part and merge successive color subclasses during RLE encoding. Further processing is unchanged. As a result we obtain the same segmentation results as depicted in Fig. 2, but the processing time can be even 50% shorter (compare Table 1). Experiments show that although the final execution time is longer than for a single pass of the original method, using up to five thresholds still enables real-time (below 40 ms) performance. Moreover, since usually three thresholds are enough, there is still some time left for additional calculations between the image frames.

The main disadvantage of our proposal is that the original limit of 32 different colors is further narrowed proportionally to the number of thresholds for a single

color (five thresholds per color allow simultaneous segmentation of 6 different colors only, but three thresholds make it 10). That original limit, however, can be easily doubled if 64-bit integers are used instead of 32-bit integers in the current version.

**Table 1.** Average execution time of the original method and our modification for different number of thresholds. The performance tests were conducted on a set of over 50 images containing a Rubik's Cube on a 2,13 GHz PentiumM PC. Results may vary depending on the nature of images (amount of noise, number and area of the segmented regions, etc.).

Number of thresholds	Original method [ms]	Our modified method [ms]	Performance gain [%]
1	16	16	0
2	33	22	33
3	49	27	45
4	63	33	48
5	81	39	52

#### 4 6-DOF Visual Servoing as a Benchmark for the Proposed Modification

In order to test the advantages of our modification in an application we decided to use it as a part of feature extraction routine in a visual servoing task [6]. We use a Rubik's Cube as an object for visual identification, localization and tracking, but any object with at least four distinctive planar features would be acceptable [4].

The entire servoing process (position-based with a stand-alone camera [5]) is realized within the MRROC++ programming framework [7]. The MRROC++ is a layered structure of the following processes:

- Effector Driver Process (EDP) - computes inverse kinematics and controls joints in configuration space,
- Virtual Sensor Process (VSP) - retrieves information from a sensor and makes it useful for high level control,
- Effector Control Process (ECP) - responsible for high level control in the task space,
- Master Process (MP) - effective when more than one effector is used to coordinate them
- User Interface.

The centroids of four corner tiles of one of the cube's faces form image features  $f_g$ . Since they are coplanar, the pose of the cube  $G$  with respect to the camera coordinate frame  $C$  can be estimated ( ${}^C T_g$ ). To do it properly camera's intrinsic (focal length, principal point and optical distortion) and extrinsic parameters (the camera's pose in respect to robot coordinate frame  ${}^0 T_G$ ) should be computed beforehand [1][3]. This pose is transformed into the robot coordinate frame 0, thus  ${}^0 T_G$  is obtained. The ECP receives from the VSP the  ${}^0 T_G$  and the current end-effector pose  ${}^0 T_E$  from the EDP computing the direct kinematics task. In consequence the control error  $\varepsilon$  ( ${}^E T_G$ ) can be

computed. The macrostep generator produces a motion trajectory  $M$  using the control error. The trajectory  $M$  is a set of macrosteps, i.e., a sequence of poses  ${}^0T_{E'}$  through which the end-effector is to pass. Only the first pose of  $M$ , i.e.,  ${}^0T_{E'}$ , is sent to the EDP. The step generator within the EDP divides the macrostep into even smaller steps, each representing a pose. The calculated pose is transformed by the inverse kinematics procedure into joint coordinates  $\Theta_d$ .  $\Theta_d$  is used as the desired value for the regulator producing  $\Theta_u$ , i.e., PWM ratio. While the ECP executes the steps of the macrostep, the ECP computes a new trajectory based on the new information delivered by the VSP which makes smooth movement.

The macrostep generator is based on proportional controller working decoupled for each pose component (3 translations and 3 rotations). A gain for each component is set empirically to filter image noise (the regulator works as a lowpass filter).

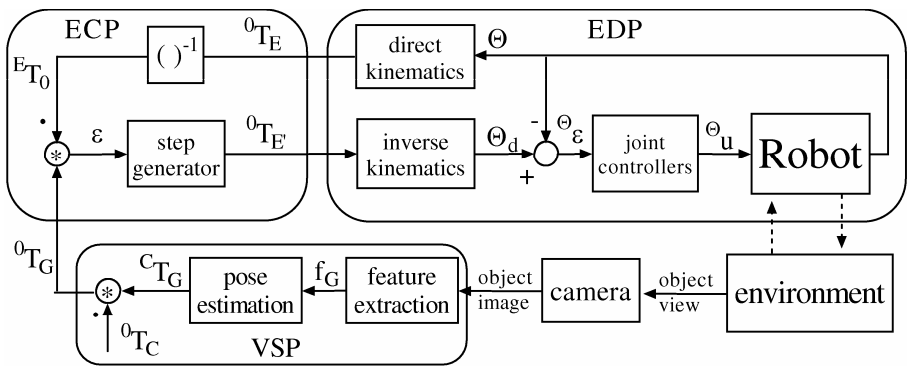
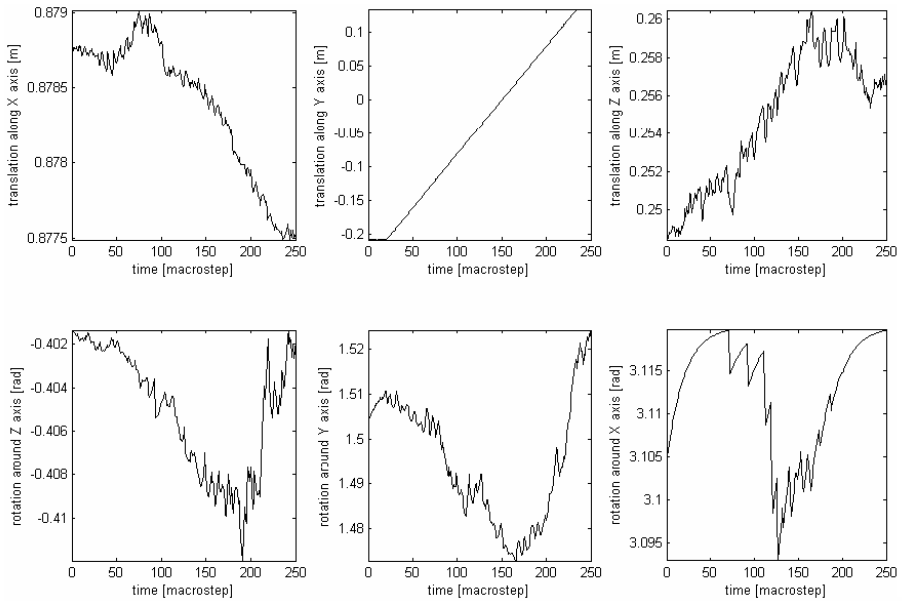


Fig. 3. The structure of the MROCC++ control system

One of the crucial aspects of visual servoing is precise localization of image features. Although they are calculated with subpixel precision, the influence of improper segmentation can lead to errors measured in pixels (the bigger the object the bigger the error). Our features (the cube’s tiles) fit a square of approximately 15-20 pixels side and the error introduced by false segmentation could be as high 2 pixels. This causes serious localization errors. After introducing our modification this error does not exceed a single pixel so the localization noise propagated to the servocontroller is minimal. The vision system is able to determine the Rubik’s Cube position and orientation in space in respect to the camera (1.5 meters apart) with the following precision:

- position in the plane perpendicular to the camera axis:  $\pm 1$  mm
- position along the camera axis:  $\pm 5$  mm
- orientation around axes perpendicular to the camera axis:  $\pm 3^\circ$
- orientation around the camera axis:  $< \pm 1^\circ$ .

After introduction of our modification the visual servoing system runs much smoother as it receives 6-DOF localization information with much higher accuracy without compromising the frequency (25 Hz).



**Fig. 4.** Trajectories of a robot manipulator tracking the Cube traveling horizontally with a constant velocity

## 5 Summary

We have presented a modification of a fast color image segmentation algorithm that makes the segmentation process much more robust to color variations and provides results similar to adaptive thresholding. We also showed that it is possible to realize it in real-time on a 2,13 GHz PentiumM based PC when up to five thresholds are defined (in most cases just three thresholds are enough).

The proposed modification can be widely used in all the applications that require fast but precise color segmentation. The accuracy of the method, which is inversely but not linearly proportional to its execution time, can be tuned by selecting the required number of thresholds.

The proposed modification has been successfully implemented in a visual servoing task, which ran smoother due to real-time visual feature localization with increased precision.

## References

1. Bouguet, J.: Camera Calibration Toolbox for Matlab, [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)
2. Bruce, J., Balch, T., Veloso, M.: Fast and Inexpensive Color Image Segmentation for Interactive Robots, Proc. IROS-2000.



3. Heikkla, J., Silvén, O.: A four-step camera calibration procedure with implicit image correction. In: 1997 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, (1997) , 1106–1113
4. Horaud, R: New methods for matching 3D objects with single perspective view. IEEE Trans. on Pattern Analysis and Machine Intelligence, (1987), 9(3):401–412
5. Hutchinson, S. A., Hager, G. D., Corke, P. I.: A tutorial on visual servo control. IEEE Trans. on Robotics and Automation, (1996), 12(5): 651–670
6. Zieliński, C., Szykiewicz W., Winiarski T., Staniak M.: Rubik's Cube Puzzle as a Benchmark for Service Robots - MMAR 2006
7. Zieliński, C.: The MRROC++ system. In 1st Workshop on Robot Motion and Control, RoMoCo'99, Kiekrz, Poland, (1999) 147–152

# A Neural Framework for Robot Motor Learning Based on Memory Consolidation

Heni Ben Amor<sup>1</sup>, Shuhei Ikemoto<sup>2</sup>, Takashi Minato<sup>2</sup>, Bernhard Jung<sup>1</sup>,  
and Hiroshi Ishiguro<sup>2</sup>

<sup>1</sup> VR and Multimedia Group, TU Bergakademie Freiberg, Freiberg, Germany  
{amor, jung}@informatik.tu-freiberg.de

<sup>2</sup> Department of Adaptive Machine Systems, Osaka University, Osaka, Japan  
{ikemoto, minato, ishiguro}@ed.ams.eng.osaka-u.ac.jp

**Abstract.** Neural networks are a popular technique for learning the adaptive control of non-linear plants. When applied to the complex control of android robots, however, they suffer from serious limitations such as the moving target problem, i.e. the interference between old and newly learned knowledge. However, in order to achieve lifelong learning, it is important that robots are able to acquire new motor skills without forgetting previously learned ones. To overcome these problems, we propose a new framework for motor learning, which is based on consolidation. The framework contains a new rehearsal algorithm for retaining previously acquired knowledge and a growing neural network. In experiments, the framework was successfully applied to an artificial benchmark problem and a real-world android robot.

## 1 Introduction

Neural networks have proved to be powerful and popular tools for learning motor control of robots with many degrees of freedom. However, some critics arouse about the use of sigmoidal neural networks (neural networks with sigmoidal kernel) for robot control. For example, Vijayakumar and colleagues [1] identified the following drawbacks of sigmoidal neural networks:

1. The need for careful network structure
2. The problem of catastrophic forgetting
3. The need for complex off-line retraining

The first point refers to the complex relationship between the network structure and its ability to learn particular functions. It is well known in the machine learning community that particular functions (for instance XOR) cannot be learned if the network structure is not appropriately chosen. However, pre-defining the network structure also means limiting the amount of learnable data. The problem of catastrophic forgetting refers to the interference between previously and newly learned knowledge. When been sequentially trained on two problems  $A$  and  $B$ , it is often found that the network will have forgotten most of its knowledge of problem  $A$  after training on problem  $B$ . However, if we really

want to achieve lifelong learning robots [10], it is important that such robots are able to acquire new skills without forgetting previously learned ones. According to Vijayakumar et al. [11] complex off-line learning is needed to overcome the problem of catastrophic forgetting, which “from practical point, (this approach) is hardly useful”. In this paper, we propose a new framework inspired by the architecture of the human brain and the process of *consolidation*. Main components of this framework are two neural networks, representing the short- and longtime memory respectively. In order to avoid that the longtime memory forgets any precious information, we employ cascade-correlation learning [1] and an improved rehearsal mechanism. Cascade-correlation adapts the network structure according to the problem difficulty, while rehearsal makes sure that old knowledge is retained.

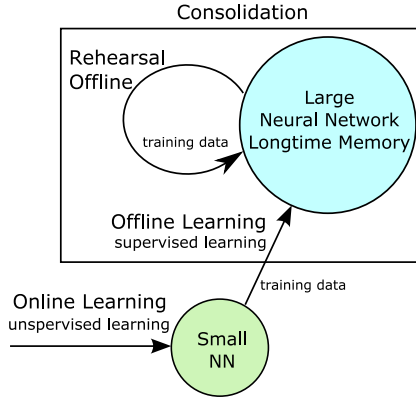
## 2 Consolidation Learning

All problems of sigmoidal neural networks mentioned in the introduction can interestingly also be found in humans. In [6] it was demonstrated that two motor tasks may be learned and retained, only if the training sessions are separated by an interval of approximately 6 hours. Otherwise, learning the second task leads to an unlearning of the internal model of the first task. However, if sufficient temporal distance is taken between the tasks, then the corresponding motor skills are retained for a long time (at least 5 months after training). The reason for this is a process called *consolidation*. During training, the acquired information is first stored in the prefrontal regions of the cortex. In this phase the information is still fragile and susceptible to behavioral interference. Then, approx. 6 hours after training, consolidation shifts the information to the premotor, posterior and cerebellar cortex in which the information is stored for a long period of time without being disrupted. In this process the brain engages new regions to perform the task at hand.

### 2.1 A Computational Framework for Consolidation Learning

In this section we propose a new computational framework for motor learning based on consolidation. Figure 1 shows the components and the working of this new framework.

An important feature of the proposed consolidation framework is the separation between short-time memory (represented by the small neural network) and the long-time memory (represented by the large neural network) of the system. Each new task or subtask is first learned **online** by the small neural network. Learning is continued until the network becomes an expert in that particular task. Once this online phase is finished the expert network teaches newly acquired knowledge to the large neural network in the consolidation phase. This process is done **offline**. To ensure that the network has enough memory capacity to learn the new knowledge, learning is performed using cascade-correlation. At the same time, old knowledge contained in the large neural network is rehearsed in order to avoid catastrophic forgetting.



**Fig. 1.** Consolidation learning framework. A new problem is first learned by a small neural network. The information gained from this NN is then used to teach a large neural network in an offline process involving rehearsal of old information.

---

**Algorithm 1.** Pseudo-code for consolidation learning  $i = 0$

---

- 1: Learn new task  $t_i$  unsupervised learning (online)
  - 2: Create set  $A$  with  $n * \beta$  supervised data from small neural network
  - 3: Create set  $B$  with  $n * (1 - \beta)$  supervised data from large neural network
  - 4: **while**  $e_n > desired\_error$  **do**
  - 5:   Learn  $A$  and rehearse  $B$
  - 6:   Compute network error  $e_n$
  - 7: **end while**
  - 8:  $i \leftarrow i + 1$
  - 9: GOTO 1
- 

Algorithm 1 illustrates the coarse flow of consolidation learning. Here, the set  $T = \{t_1, \dots, t_k\}$  refers to the  $k$  subtasks to be sequentially learned. The number  $\beta$  is a biasing parameter, which biases learning towards new or towards old information. If  $\beta$  equals zero, then the network will mainly try to remember old knowledge without learning new information. Empirically we found out that setting this value to 0.5 results in a good tradeoff. Finally, the number  $n$  represents the size of the training set to be used during consolidation.

A critical point in this algorithm is the reinstatement, i.e. the creation of supervised data used for rehearsal. One way to create the supervised data would be to store the old input vectors which were previously used for learning. By computing the output of the large neural network for each of these input vectors, we get input-output pairs which can be used as a training set during rehearsal. The drawback of this approach is that it requires storing input data of all training

phases so far. Another approach to reinstatement, called “pseudorehearsal”, is to use random input vectors when computing the rehearsal training set from the large neural network. This has the advantage of getting rid of additional data to be saved. On the other hand, it might deteriorate the performance of consolidation. A random input might lie in an area of the input space, for which the long-time memory did not get any training data so far. In such a case, the rehearsed pattern might conflict with a newly learned pattern from the small neural network. When using growing techniques, such a conflict will result in more and more expert neurons being added to the network. The consequence is increasing error, learning time and network complexity. The following rehearsal algorithm aims at avoiding such situations.

## 2.2 Stimulation Based Pseudorehearsal

Previous rehearsal and pseudorehearsal techniques create a buffer which combines a number of new (from set  $A$ ) and old patterns (from set  $B$ ). Training the neural network with this buffer ensures that new information is learned and old information retained. In contrast to this, our new rehearsal algorithm called *stimulation based pseudorehearsal* explicitly distinguishes between a ‘pure’ rehearsal step and a step for learning new knowledge. These steps are executed alternately. Learning is executed until either the network error falls below the desired error or until convergence. In the latter case the problem is not successfully solved and thus new neurons will be trained and recruited to the network. The neurons are trained to maximize correlation between their output and the set  $A$  of new patterns. New neurons which are added to the network should act as experts for the newly acquired knowledge solely while old expert neurons represent previously acquired knowledge. These expert neurons are stimulated every second epoch, so they don’t forget their information. After insertion of a new neuron into the network, rehearsal and learning are repeated. Algorithm 2 shows a pseudo-code implementation of this rehearsal technique.

---

### Algorithm 2. Pseudo-code for stimulation based rehearsal

---

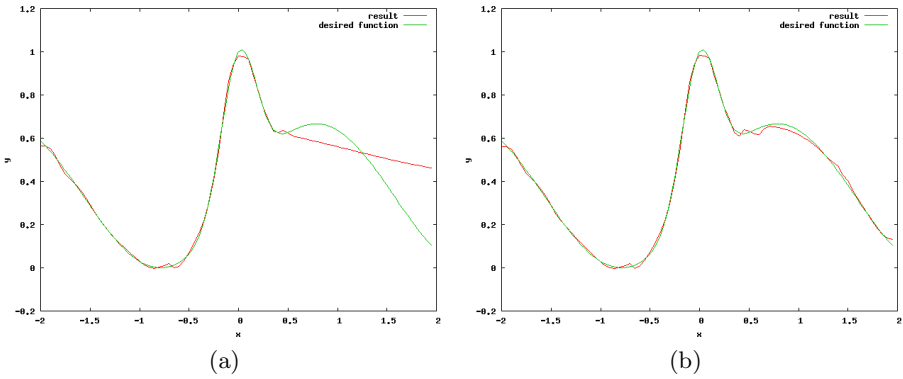
```

1: Learn  $B$  for one epoch
2: Learn  $A$  for one epoch and get network error  $e_n$ 
3: if  $e_n < \text{desired\_error}$  then
4:   Learning finished
5: else if not converged then
6:   GOTO 1
7: else
8:   Create new neuron and maximize correlation with  $A$ 
9:   Add new neuron to network
10:  GOTO 1
11: end if

```

---

In order to verify the strength of this pseudorehearsal technique, we conducted a set of experiments. The experiments were performed using both the original pseudorehearsal algorithm (for short: PR1) as described in [4] and stimulation based pseudorehearsal (SBPR). As benchmark we chose a problem which was already used in [5] and at the time could not be solved by a 3-layer sigmoidal feedforward neural network. The goal is to approximate the function  $y = \sin(2x) + 2\exp(-16x^2) + N(0, 0.16)$  in two steps. In the first step a set of 130 data points which are uniformly distributed around in  $x \in [-2.0, 0.5]$  are presented to the network. After learning these data points, the network is then presented with a new set of 70 new data points in  $x \in [0.5, 2.0]$ . It was shown that a neural network would forget the knowledge about the first part of the function after it has learned the second set of data points.



**Fig. 2.** Two phases of consolidation learning with stimulation based pseudorehearsal: the function  $y = \sin(2x) + 2\exp(-16x^2) + N(0, 0.16)$  (green) is to be learned in two steps. In Figure (a) the first part of the function was approximated. In Figure (b) the second part was approximated, while information of the first part was rehearsed.

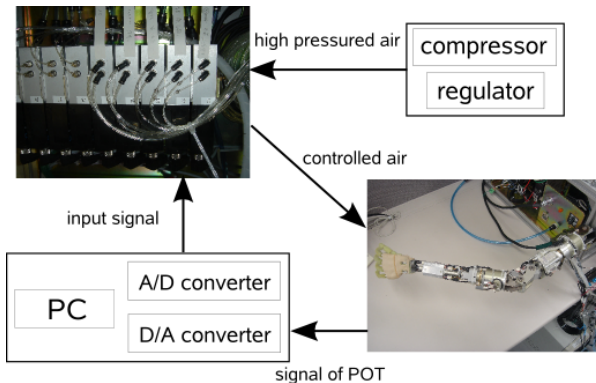
Figure 2(a) shows the result of the training after learning the first set of data points. The neural network successfully learned to approximate the function in the range  $x \in [-2.0, 0.5]$ . In the next step the network was provided with points from  $x \in [0.5, 2.0]$  and SBPR was used in order to make sure that new knowledge is learned without forgetting the old. Figure 2(b) clearly shows, that the problem was successfully solved. Table 1 presents a comparison between result of PR1 and SBPR. From the table we can see, that the new variant leads to lower network error and network size. The mean squared error is about 0.0024 with SBPR, while it is approximately 6 times as high with PR1. The average number of neurons with SBPR is only 12.9 neurons compared to 26.4 neurons with PR1. In our experiments we also found out that on average the SBPR algorithm runs three times as fast as PR1.

**Table 1.** Performance of the original pseudorehearsal algorithm (PR1) and the stimulation based pseudorehearsal algorithm (SBPR) on an incremental approximation problem

Method	Mean Sq. Error	Std. Dev. Error	Average Num. Neurons	Std. Dev. Neurons
PR1	0.0147	0.0199	26.4	11.5
SBPR	<b>0.0024</b>	<b>0.0027</b>	<b>12.9</b>	<b>2.8</b>

### 3 Experiments and Result

We evaluated consolidation learning on the pneumatic arm of an android robot called Repliee Q2. The android system adopts small pneumatic actuators and long, thin air tubes to realize a compact mechanism because the android system must have a very humanlike appearance. This results in strong non-linearity and provides a large response lag, including a large dead time. Controlling the motion of such a system can be difficult. One well-known method for learning desired motion is feedback error learning (FEL) [2]. However, it has been reported that such a large dead time makes feedback control stabilization more difficult [3]. This may disturb the feedback error learning applied to the motor learning of the android. Another learning technique which was shown to achieve state-of-the-art results in control tasks is Neuro Evolution on Augmenting Topologies (NEAT) [8]. NEAT uses a genetic algorithm to evolve appropriate neural networks. In our evaluation consolidation learning is applied to the control of a one-DoF air driven manipulator. Three different reference trajectories were sequentially used for learning. The robot arm had to learn to follow these reference trajectories by minimizing the error. After sequentially learning the three reference trajectories, the resulting neural network was tested for generalization and forgetting. For this, a validation set was created which included the three reference trajectories and two new (not learned) trajectories. The tests were performed using FEL,

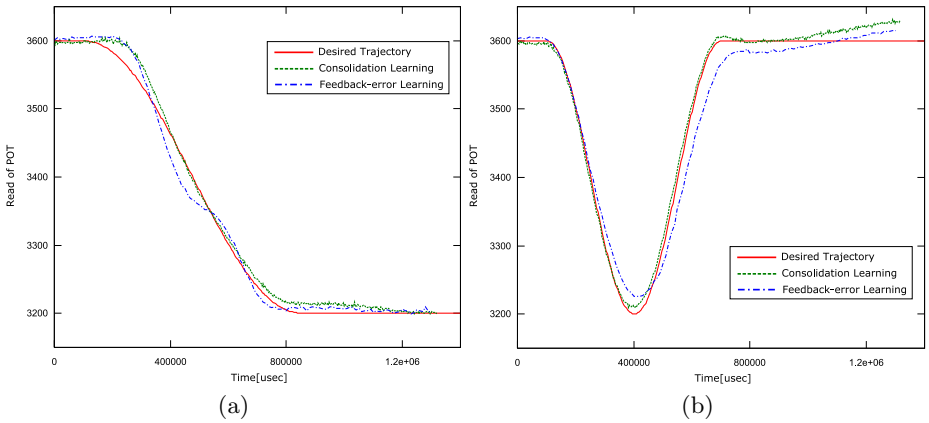


**Fig. 3.** The setting of the experiment for testing consolidation: An android arm is actuated by air pressure coming from a compressor

NEAT and consolidation learning. The dimension of the input vectors was 31. For FEL the number of hidden neurons was set to 20.

### 3.1 Results

In Figure 4 we see two out of five validation trajectories (red), and the tracking performed by consolidation learning (green) and by FEL (blue). Trajectory (a) was also used for learning, while trajectory (b) was only used for validation. It can be seen that the network learned with consolidation learning was able to follow both trajectories. The network was also able to solve both trajectories that have been seen and new trajectories. In contrast to this, the tracking achieved by FEL did not appropriately follow the reference trajectories; e.g it oscillated around trajectory (a), which lead to unnatural movements and friction between the robot parts. We also computed the mean squared error performed by the different learning algorithms in this setting. Consolidation learning performed best followed by NEAT and then FEL with a mean squared error of **0.044**, **0.055** and **0.056** respectively.



**Fig. 4.** Different desired and executed trajectories of the android arm after consolidation learning is finished. Time is given in microseconds. The trajectory (a) was also trained with, while trajectory (b) was only used in the validation phase.

## 4 Related Work

In [4] Robins discussed different methods for rehearsal and proposed the “pseudorehearsal” approach, which is also adopted in this paper. Tani [9] showed that a consolidation can help to reduce catastrophic forgetting in recurrent neural networks. In his approach, however, he uses a database to store each seen input pattern. Depending on the application, this can lead to high memory consumption. In [7] Poirier and Silver present a framework for Multiple Task Learning (MTL) based on ideas from consolidation. Although this approach shares some concepts



with our work, it is mainly focused on the MTL domain. Because backpropagation learning is used, a great deal of effort has to be put into the design of network architecture. Additionally, this means that the amount of learnable knowledge is limited. In contrast to this, the neural networks in our paper scale up to the amount of knowledge to be learned, by inserting new neurons when needed.

## 5 Conclusion

In this paper we presented a new computational framework for robot motor learning based on a process called consolidation. Consolidation is inspired by neurobiological findings in humans and animals. The framework was evaluated on an android robot and found to be successful in avoiding catastrophic forgetting and achieving high generalization. Furthermore, it removes the need for specifying the network structure prior to learning. For the future we aim at investigating how well this framework scales up to larger problems and how many times learning can be performed without deteriorating knowledge which was learned at the beginning. Another interesting point for further investigation, is the comparison of the influence of different growing techniques on the performance of our framework. Finally, we aim at applying the framework to a more sophisticated, new android robot and to simulated humans in virtual reality environments.

## References

1. S. Fahlman and C. Lebiere. The cascade-correlation learning architecture. Technical Report CMU-CS-90-100, Carnegie Mellon, University, Pittsburgh, 1991.
2. M. Kawato. Feedback-error-learning neural network for supervised motor learning. *Advanced Neural Computers*, pages 365–472, 1990.
3. Y. Li and T. Asakura. Occurrence of trajectory chaos and its stabilizing control due to dead time of a pneumatic manipulator. *JSME International Journal Series C*, 48(4):640–648, July 2005.
4. A. Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connect. Sci.*, 7(2):123–146, 1995.
5. S. Schaal and C. G. Atkeson. Constructive incremental learning from only local information. *Neural Comput.*, 10(9):2047–2084, 1998.
6. R. Shadmehr and H. Holcomb. Neural correlates of motor memory consolidation. *Science*, 277:821–825, 1997.
7. D. L. Silver and R. Poirier. Sequential consolidation of learned task knowledge. In *Canadian Conference on AI*, pages 217–232, 2004.
8. K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002.
9. J. Tani. An interpretation of the 'self' from the dynamical systems perspective: A constructivist approach. *Journal of Consciousness Studies*, 5(5-6), 1998.
10. S. Thrun. *Lifelong Learning Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.
11. S. Vijayakumar and S. Schaal. Fast and efficient incremental learning for high-dimensional movement systems. In *Proc. IEEE Int'l Conf. Robotics and Automation*, pages 1894–1899. IEEE Press, Piscataway, N.J., 2000.

# Progressive Optimisation of Organised Colonies of Ants for Robot Navigation: An Inspiration from Nature

Tatiana Tambouratzis

Department of Industrial Management & Technology, University of Piraeus,  
107 Deligiorgi St, Piraeus 185 34, Greece  
tatianatambouratzis@gmail.com  
<http://www.tex.unipi.gr/dep/tambouratzis/main.htm>

**Abstract.** This piece of research introduces POOCA (Progressive Optimisation of Organised Colonies of Ants) as an appealing variant of the established ACO (Ant Colony Optimisation) algorithm. The novelty of POOCA lies on the combination of the co-operation inherent in ACO with the spread of activation around the winner node during SOM (Self-Organising Map) training. The principles and operation of POOCA are demonstrated on examples from robot navigation in unknown environments cluttered with obstacles: efficient navigation and obstacle avoidance are demonstrated via the construction of short and – at the same time - smooth paths (i.e. optimal, or near-optimal solutions); furthermore, path convergence is speedily accomplished with restricted numbers of ants in the colony. The aim of this presentation is to put forward the application of POOCA to combinatorial optimisation problems such as the traveling salesman, scheduling etc.

## 1 Introduction

Computational intelligence draws its inspiration from

- the brain and the nervous system
- the social behaviour demonstrated by ensembles

of living beings. An illustrative example of the latter is real ant colonies, where the sum of the parts is clearly more than the parts themselves: although ants acting as individuals are unable to successfully navigate or to collect food, when working cooperatively, they are capable both of locating food in extensive and intricate environments and of finding “shortest” trails from their nest to the foodsource(s) and back while avoiding obstacles [1]. Ants (e.g. the Pharaoh species) forage their environment by moving along a multitude of fork-junctions; such a junction appears whenever a trail is divided into two sub-trails, which diverge at an angle of around  $53^\circ$  from each other [2-3]. Hence, although these trails may not be shortest in the Euclidean sense, they constitute the shortest combination of sub-trails along the fork-junctions.

Ant colony optimisation (ACO) constitutes a part of the larger field of swarm intelligence (SI), that is an assortment of nature-inspired general-purpose multi-agent

stochastic search heuristic procedures in which the behavioural pattern of ants, bees, termites and other social insects is investigated and simulated in an effort to provide adequately good solutions to NP-hard combinatorial optimisation problems in a reasonable amount of time. ACO – including the ACO meta-heuristic and the related ACO algorithms - has been introduced, detailed and developed extensively in the literature ([4-5], for a survey the reader is referred to [6-7]).

In this piece of research POOCA (Progressive Optimisation of Organised Colonies of Ants) is put forward as an appealing variant of the established ACO algorithms. POOCA combines the co-operation inherent in ACO with the spread of activation around the winner node during SOM (Self-Organising Map [8]) training. The principles and operation of POOCA are demonstrated on examples of robot navigation in unknown environments cluttered with obstacles: efficient navigation and obstacle avoidance are demonstrated via the construction of short and – at the same time - smooth paths (i.e. optimal, or near-optimal solutions); furthermore, a restricted number of ants in the colony is sufficient for rapid path convergence. The presented POOCA characteristics render the methodology interesting for combinatorial optimisation problems such as the traveling salesman, scheduling etc.

## 2 Real Ant Colonies

In most<sup>1</sup> real ant colonies, food discovery as well as path formation are accomplished via the laying of the chemical substance pheromone. Pheromone constitutes an indirect<sup>2</sup> but extensive mechanism of communication and reinforcement which acts as a highly robust as well as adaptive form of positive-feedback and allows the ant colony to accomplish complex tasks that are beyond the power of the individual. Pheromone works as:

- A trail marker of the individual ant. By being deposited in small amounts on the ground by each ant traveling in the environment in search of food, pheromone trails originating from the nest are formed. Pheromone evaporates with time (e.g. for the *Lasius Americanus* species evaporation occurs in about 2 minutes, the time it takes to cover about 40cm), whereby more recent parts of the trails are better remembered than older, decaying parts which are eventually forgotten.
- An indirect, collective and interactive memory for all the ants in the colony (stigmetry [9]): each ant can smell the pheromone deposited by all the ants in the colony and - in the effort to reach the foodsource - it prefers (in a probabilistic sense) to follow heavily marked (i.e. frequently visited) pheromone trails rather than to investigate unvisited or forgotten parts of the environment; such a preference further reinforces pheromone-rich trails. Still, the occasional selection of lightly marked or unmarked/decayed parts of the environment allows the

---

<sup>1</sup> Not all ant species rely on continual pheromone laying for path creation (as do the Argentine ants); for instance, ants of the Sahara and Namib deserts use no chemical markers, while the *Tetramorium Caespitum* ants lay pheromones only on the way back from successful nest-foodsource trips.

<sup>2</sup> Population-wise global and – at the same time - space-wise local: although laid and sensed by the entire ant colony, pheromone can only be deposited and detected near the ant location.

discovery of alternative trails and, thus, successful navigation either when the environment changes (e.g. obstacles appear/disappear/move, foodsources become depleted) or when trails are partially destroyed.

- A unique trail attractor for the colony: at the beginning of navigation ant movement is random, whereby a multitude of trails are formed. Shorter trails connecting the nest to the foodsource become richer in pheromone faster than longer trails, whereby the probability that the ants will select these trails increases; this – in turn – further reinforces the accumulation of pheromone (the differential length effect). Eventually, the entire colony converges to the shortest trail between the nest and the foodsource. In cases where multiple foodsources exist, the shortest trail connecting the nest with the nearest foodsource is settled upon first; this trail decays after the foodsource is exhausted; acting as before, the foraging ants converge upon the shortest trail to the second nearest foodsource until this is also exhausted and so on. In all, the colony is capable of finding the shortest trail from the nest to each of the foodsources, where the order of trail creation is defined by the proximity of the foodsources to the nest.

### 3 ACO

The operation of real ant colonies is simulated by ACO with modifications that render the methodology computationally accurate and efficient: time and space are made discrete, memory is employed in order for the ants to be able to remember past actions<sup>3</sup>, pheromone is initialised to a small positive value rather than to zero, a differential amount of pheromone is deposited at different locations or time-steps, backtracking and other artificial-intelligence-inspired procedures are employed etc.

ACO is based upon the following:

- transformation of the problem into a graph-like structure,
- problem-dependent formulation of the distances between connected nodes of the graph,
- construction and updating of a sequence of nodes (corresponding to a pheromone trail), implemented via the following processes: seek-the-foodsource, collect-food, return-to-nest, deposit-food-to-nest, create/select-trail etc.

Owing to the fact that paths and path-length-based search can be appropriately adapted to represent almost any combinatorial problem, ACO enjoys a wide range of successful applications (either as an independent methodology or as a combination of ACO with genetic algorithms, TABU search, local search etc.) including but not limited to (for a more thorough bibliography and a comparison with other soft computing techniques the reader is referred to [4-5,7]):

- static problems, e.g. the symmetric and asymmetric traveling salesman, quadratic assignment, job-shop and other forms of scheduling, resource allocation, communication, mobile etc. network assignment, circuit design, sequential

---

<sup>3</sup> For instance, the trail is updated off-line (once the entire nest-to-foodsource or foodsource-to-nest trail has been completed) whereby memory of past actions is required, rather than on-line (at each time-step) in which case no memory is employed.

ordering, graph colouring, maximum clique, minimum spanning tree, vehicle routing, bandwidth minimisation, bioinformatics, and

- dynamic problems, e.g. connection-oriented and connection-less network routing, learning automata and game playing, dynamic graph colouring, data mining.

## 4 POOCA

POOCA, combines the traditional ACO characteristics with the spread of activation around the winner node during SOM training: concurrently with the modification of the weight of the winner node after presentation of a training pattern to the SOM, the weights of the neighbouring nodes are also modified to smaller degrees. The spread of activation is initially applied to a relatively large neighbourhood around the winner node; the neighbourhood shrinks as training progresses and, at the end of training, becomes confined to the winner node. This weight modification technique ensures that – after training – the weights of the nodes change gradually along the SOM, whereby smoothly varying neighbourhoods of nodes (topographical organisation) are created.

The main POOCA characteristics are detailed next for robot navigation in unknown environments cluttered with obstacles. This problem has been selected due to its ease of visualisation; furthermore, the fact that only a fraction of the environment constitutes the solution (navigation path) facilitates illustration.

### 4.1 Robot Navigation

Recently, the control of automated guided vehicles (AGVs) has received considerable attention in industrial manufacturing systems. AGVs can be made to follow pre-wired or otherwise predefined paths between a desired start (e.g. unloading) point and one or more end (e.g. loading) points in an environment cluttered with obstacles; alternatively, a path can be planned in real-time according to the constraints and requirements of the task at hand. Both versions of the path planning problem are NP-complete, as is a simpler problem of planning the motion of a point robot among 3-D polyhedral obstacles [10]. To make matters worse, the environment is often dynamic in the sense that some of the obstacles may appear/disappear/move and/or end-point(s) may become unusable (e.g. when the loading operation has been completed).

### 4.2 ACO and Robot Navigation

ACO has been employed as a tool for navigation of AGVs e.g. [11-12]. However, communication and information exchange are by no means local in these approaches: either the information acquired by each AGV (concerning locations of interest within the environment) is communicated over a radio-network to all AGVs as soon as it is acquired, or a partial – and common to all AGVs – model of the environment is constructed to complement the local ACO model.

### 4.3 POOCA and Robot Navigation

In POOCA, the navigation environment has been discretised:

- In time; ant motion is performed in terms of time-steps, synchronously for all ants.
- In space; the environment is represented as a rectangular grid where each cell stands for a point of the environment, be it a transversable (start point, end point, free space<sup>4</sup>) or non-transversable (part of an obstacle<sup>5</sup>) ant position. An ant transition constitutes a move from one transversable cell of the grid to an adjacent transversable cell.

Knowledge of the environment is initialised with the location of the start-cell only; the location of the end-cell remains unknown until it is first reached by an ant. The same is true of the location, size and shape of the obstacles: these are initially unknown and become incorporated in the environment only once (and if) an ant of the colony accidentally bumps into them; even then, only the particular cell of the obstacle is identified, i.e. the obstacle is only partly revealed. Hence, knowledge of the environment is gradual and may remain incomplete even after the end of POOCA operation.

Each cell of the ant-grid is initialised with a small positive pheromone value that is common to all cells. At the first time-step, all ants are placed on the start-cell and their start-to-end trips are initialised with the start-cell. At every subsequent time-step, the next cell to be visited by each ant is selected among the cells which are adjacent to the cell currently occupied by the ant and which have not already been visited during the current trip of the ant; roulette-wheel selection has been implemented as a pseudorandom proportional<sup>6</sup> process expressing the preference for cells with higher pheromone values over those with lower pheromone values

$$prob^{adjacent\ cell_i} = \frac{pheromone(adjacent\ cell_i)}{\sum_j pheromone(adjacent\ cell_j)} \quad (1)$$

where  $prob^{adjacent\ cell_i}$  is the probability of selecting  $adjacent\ cell_i$  for ant transition,  $pheromone(adjacent\ cell_*)$  is the pheromone value of cell  $*(=i, j)$  and  $j$  ranges over all adjacent cells<sup>7</sup>. If the selected cell constitutes (part of) an obstacle, its pheromone value is set to zero for the remainder of POOCA operation and roulette-wheel is repeated. If no cell can be found for ant transition (i.e. all adjacent cells constitute either obstacle-cells or part of the current trip), the ant backtracks one cell per time-step to the original cell and starts its trip afresh.

A combination of on- and off-line pheromone updating has been employed. Memory has been made use of:

---

<sup>4</sup> The start and end points occupy one cell each; the free space occupies a multitude of connected and unconnected cells.

<sup>5</sup> Each obstacle may occupy more than one connected cells; its arbitrary size and shape is represented by appropriate combinations of adjacent cells.

<sup>6</sup> The difference in pheromone has been implemented such that cells of high pheromone value have a significantly larger probability of being preferred over cells of lower values, thus closely imitating real ant operation [13].

<sup>7</sup> Obviously, adjacent cells of zero pheromone are not considered for selection.

- for the first start-to-end trip of each ant; off-line pheromone updating of all the cells comprising the trip is performed as soon as the trip has been accomplished<sup>8</sup>,
- for the cells comprising the current trip, in order (a) not to allow a cell of the trip to be revisited and (b) to enforce off-line updating of the pheromone values of the cells adjacent to the cells of the trip.

No memory has been employed for:

- pheromone updating of the cells comprising all subsequent start-to-end trips as well as all the end-to-start trips of each ant, i.e. the amount of pheromone of a cell is incremented on-line as soon as an ant visits it,
- the revealed obstacle cells, which are simply characterised by zero pheromone values.

Pheromone laying is leaky in that, each time a start-to-end or an end-to-start trip is completed, pheromone spreads to adjacent cells of the environment (a parallel to SOM training of a gradually shrinking neighbourhood around the winner node). The amount of leakage for each cell in the grid:

- is proportional to the number of cells in the trip which are adjacent to the cell; so, the increment in pheromone – due to leakage – is larger/smaller if the cell is adjacent to many/few cells of the path (e.g. on the concave/convex side of a bend), whereby path smoothness is promoted by discouraging sharp bends as well as merging multiple paths that are near each other<sup>9</sup>,
- diminishes at each subsequent trip of the ant. Initially, all cells (other than the start- and end-points as well as the known obstacle cells) of the grid are subject to pheromone leakage. Subsequently, the intensity of leakage progressively shrinks to encompass only those cells that are nearer to the completed trip. As a result, the trip is considerably wider at the beginning of POOCA operation and narrows down progressively to one cell (width).

The amounts of pheromone incrementing and evaporation remain constant during the entire POOCA operation, i.e. are time- and space-independent. Pheromone evaporation is performed at each time-step to all cells of the grid after the first ant has reached the end-cell (i.e. as soon as some cells of the grid are marked by pheromone) according to

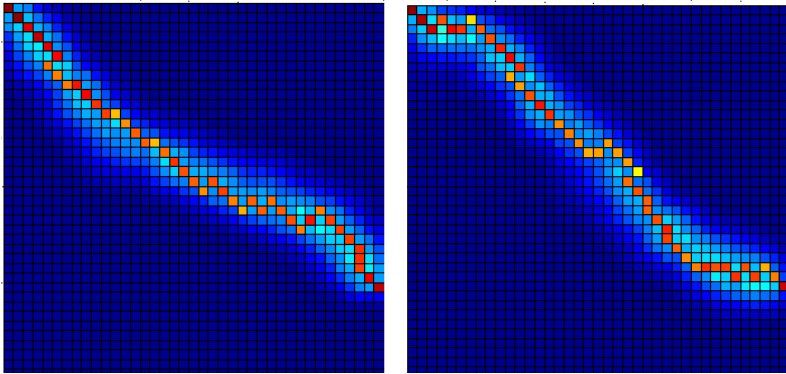
$$pheromone_{t+1}(cell) = (1 - decay)pheromone_t(cell) . \quad (2)$$

where  $t$  and  $t+1$  are subsequent time-steps and  $decay$  is a small positive number. Clearly, the amount of pheromone evaporation is proportional to the pheromone value. Finally, pheromone incrementing and evaporation are followed by pheromone normalisation in the interval  $[0,1]$  (a parallel to [14]) at each time-step following the creation of the first trip.

---

<sup>8</sup> Memory has been implemented in this case in order not to allow cyclic/oscillatory behaviour either during the ant's trip or by other ants traversing this part of the environment. Furthermore, the start-to-end trips to be created are reinforced proportionally to their time of creation (and indirectly to their length), thus simulating the differential length effect.

<sup>9</sup> Pheromone spread and path creation resemble robot "vapour trails", where the difference between left and right sensors guides towards the higher pheromone concentration.



**Fig. 1.** Path construction in an totally obstacle-free environment; three ants

## 5 Results

The navigation problems investigated here vary in size from 100 to 10000 cells arranged into rectangles of different configurations. The obstacles have been devised so as to occupy anything between 0 and 60% of the cells in the ant-grid; furthermore, obstacles have been shaped such as to form no concavities, concavities (obstacles connected into small independent groups) as well as dead-ends (maze-like connected obstacles). The removal, addition and shift of obstacles have also been studied. One start point and up to five end points have been tested.

The number of ants required for determining satisfactory paths is significantly smaller than that observed in other ACO implementations; any number between three and 15 ants have been found to create satisfactory paths, the optimal number showing a small correlation with ant-grid size. This finding is not surprising since the spread of pheromone combined with a small rate of pheromone evaporation allow the integration of the various paths created by the ants during their trips.

The spread of activation to adjacent cells of the ant-grid environment during path creation promotes the construction of short paths connecting the start and end points, as well as the circumvention of obstacles in a smooth manner. No significant oscillatory motion is apparent (compare with [15]), which is of special importance since no course-maintaining motion has been dictated (i.e. a preference for 0 over 45 over 90° etc. changes in direction); this is illustrated in Fig. 1. In fact, when no obstacles appear in the grid, trail smoothness resembles the convergence of ant trails to the straight line proposed in [16], even though no visual cues, but only pheromone concentrations, have been employed. The same is observed in long and narrow corridor-like environments [15] which have been used traditionally as a test-bed for robot navigation (Fig. 2).

In fact, it has been observed that if the ants are restricted into moving straight ahead or diverging by 60° either way (in the discrete representation of the cell-grid implemented by 0 or 45° change in direction and in an effort to create a direct simulation of real ant motion), many directional discontinuities occur.



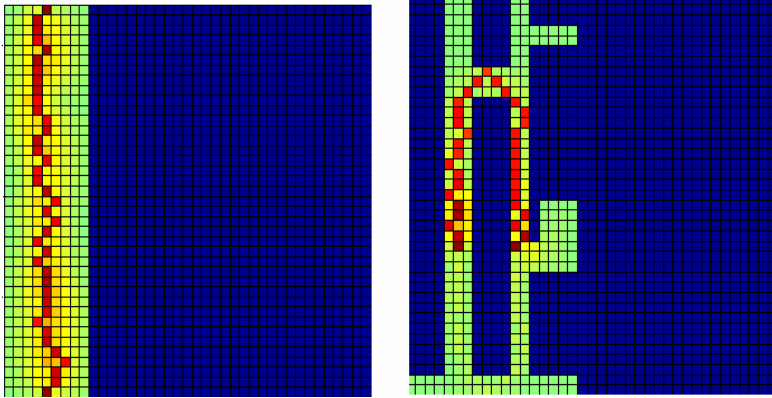


Fig. 2. construction in corridor-like environments; five ants

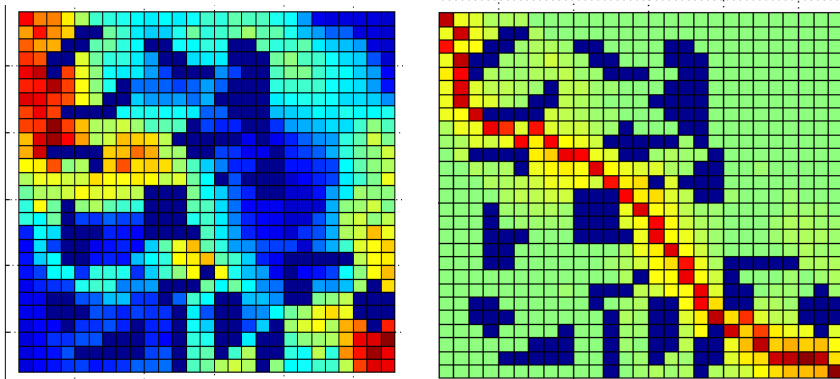


Fig. 3. Path construction in an environment cluttered with obstacles (dark collections of cells); intermediate (leftmost part) and final (rightmost part) POOCA states of operation after 25 and 250 trips, respectively; seven ants

The pheromone value of each cell epitomises the frequency with which it constitutes part of the ant trips; these trips become progressively unified so that, at the end of operation, a single narrow path is settled upon with its cells having high pheromone values (approaching 1); at the same time, all of the remaining cells have low pheromone values (approaching 0). Progressive path optimisation (shown in Fig. 3) requires that the ants complete a sufficient number of round trips; this number ranges between 100 and 250 for the problems tested here, although even as few as 25-50 trips can create satisfactory paths (the exact number depending on the size and complexity of the environment).

A variety of paths of roughly equal length are settled upon by the proposed approach (Fig. 4). Finally, as mentioned before, it is not necessary for the entire environment to be explored in order for a satisfactory path to be created: foraging has been found to be primarily focused upon the part of the environment lying between the start and end points (Fig. 5).

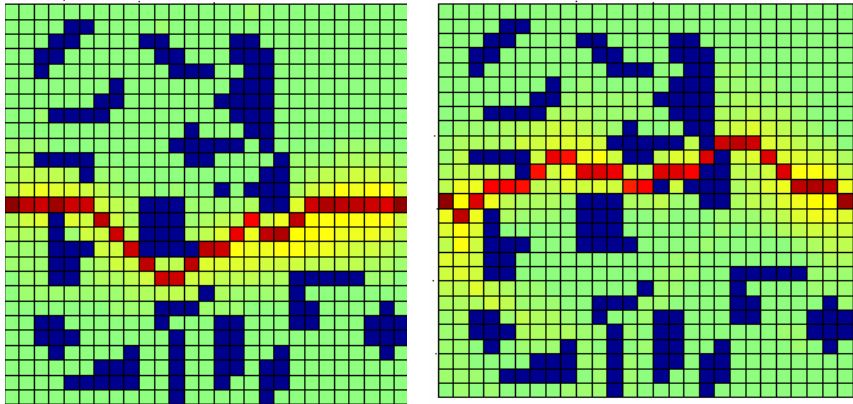


Fig. 4. Smoothness and variety of the constructed paths; five ants

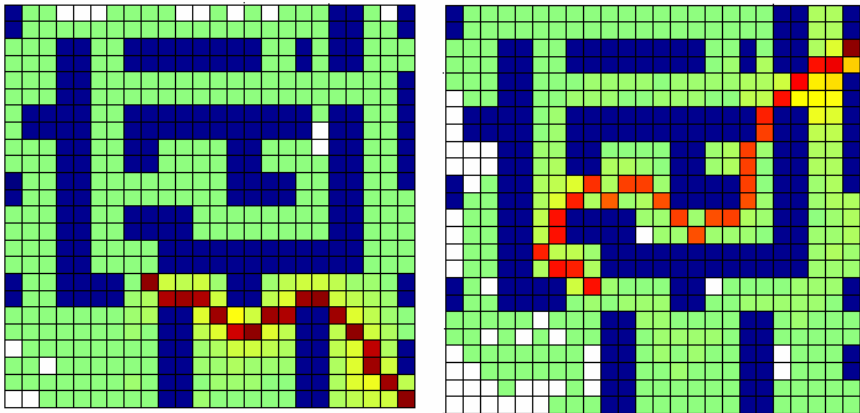


Fig. 5. Selective foraging of the environment, with unexplored (white) areas; four ants

## 6 Conclusions

POOCA has been introduced as an appealing ACO variant for robot navigation in unknown environments cluttered with obstacles. The combination of the co-operation inherent in ACO and the spreading activation around the winner node during SOM training renders POOCA capable of constructing short paths that connect the start and end points, while circumventing the obstacles in a smooth manner; furthermore, even for restricted numbers of ants in the colony, no significant oscillatory motion is observed. The observed POOCA characteristics render the methodology interesting for combinatorial optimisation problems such as the traveling salesman, scheduling etc.

## References

1. Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M.: Self-organised shortcuts in the argentine ant, *Naturwissenschaften* 76 (1989) 579-581
2. Collett, T.S., Waxman, D.: Ant navigation: reading geometrical signposts, *Current Biology Dispatch* 15 (2005) R171
3. Jackson, D.E., Holcombe, M., Ratnieks, F.L.W.: Trail geometry gives polarity to ant foraging networks, *Nature* 432 (2004) 907-909
4. Dorigo, M., Maniezzo, V., Colomi, A.: Ant system: optimisation by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics* 26 (1996) 29-41
5. Dorigo, M., di Caro, G., Gambardella, L.M.: Ant algorithms for discrete optimization, *Artificial Life* 5 (1999) 137-172
6. Blum, C.: Ant colony optimization: introduction and recent trends, *Physics of Life Reviews* 2 (2005) 353-373
7. Dorigo, M., Blum, C., Ant colony optimization theory: a survey, *Theoretical Computer Science* 344 (2005) 243-278
8. Kohonen, T.: *Self-Organising Maps – Third Extended Edition*. Springer-Verlag: Berlin, Germany (2001)
9. Grasse, P.P.: La reconstruction du nid et les coordinations interindividuelles chez *bellositermes natalensis* et *cubitermes* sp. La theorie de la stigmetrie: essai d' interpretation du comportement des termites constructeurs, *Insectes Sociaux* 6 (1959) 41-81
10. Mazer, B., Ahuactzin, J.M., Bessiere, P.: The Ariadne's clew algorithm, *Journal of Artificial Intelligence Research* 9 (1998) 295-316
11. Borisov, A., Vasilyev, A.: Learning classifier systems in autonomous agent control tasks. In: *Proceedings of 5<sup>th</sup> International Conference on Application of Fuzzy Systems and Soft Computing ("ICAFS-2002")*, Milan, Italy, September 17<sup>th</sup>-18<sup>th</sup> (2002) 36-42
12. Vaughan, R.T., Stoy, K., Sukhatme, G.S., Mataric, M.J.: LOST: localization-space trails for robot teams, *IEEE Transactions on Robotics and Automation* 8 (2002) 796-812
13. Sumpter, D.J.T., Beekman, M.: From nonlinearity to optimality: pheromone trail foraging by ants, *Animal Behaviour* 66 (2003) 273-280
14. Blum, C., Dorigo, M.: The hyper-cube framework for ant colony optimization, *IEEE Transactions on Systems, Man, and Cybernetics – Part B* 34 (2004) 1161-1172
15. Chang, C.: Using sensor habituation in mobile robots to reduce oscillatory movements in narrow corridors, *IEEE Transactions on Neural Networks* 16 (2005) 1582-1589
16. Bruckstein, A.M.: Why the ant trails look so straight and nice, *The Mathematical Intelligencer* 15 (1993) 59-62

# An Algorithm for Selecting a Group Leader in Mobile Robots Realized by Mobile Ad Hoc Networks and Object Entropy

Sang-Chul Kim

School of Computer Science, Kookmin University,  
861-1, Chongnung-dong, Songbuk-gu, Seoul, 136-702 Korea  
sckim7@kookmin.ac.kr

**Abstract.** This paper<sup>1</sup> proposes a novel algorithm for mobile robots to select a group leader and to be guided in order to perform a specific work. The concepts of mobile ad hoc network (MANET) and object entropy are adopted to design the selection of a group leader. A logical robot group is created based on the exchange of *request* and *reply* messages in a robot communication group whose organization depends on a transmission range. A group leader is selected based on the transmission of *confirmation* message from a robot who initiates to make a logical robot group. The proposed algorithm has been verified based on the computer-based simulation. The performance metric such as the number of message in order to make a logical robot group and to select a group leader is defined and verified by using the computer-based simulation.

## 1 Introduction

There has been a significant amount of research about systems of multiple mobile robots for the past decade. In particular, the cooperative aspect of such systems has been an interesting issue. Cooperation of robots refers to a situation where multiple robots operate together to perform a task that either cannot be achieved by a single robot, or whose execution can be improved by using more than one robot, thus obtaining higher performances [1]. Robot research in the past focuses on improving the performance of a single robot, however, in nowadays, it has been issued in the research of multiple mobile robots system. In particular, the cooperative aspect of the systems becomes one of main research area. Multiple robot systems can have many advantages over a single-robot system if the robots in the system cooperate in an effective and efficient manner. It is noted that there is a possibility that one robot may fail to complete its task in a timely manner due to several reasons and the other robots could be kept waiting on and on, which can cause the mission of the system incomplete [2]. It is also noted that a team of simple and cheap robots can replace a single complex and expensive one [3]. The cooperation systems are classified into four groups [3]: (1)

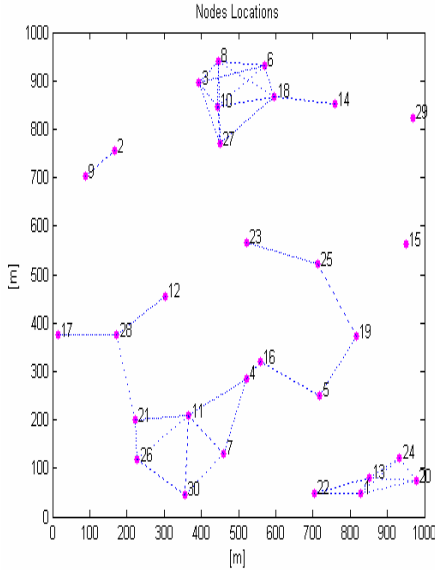
---

<sup>1</sup> This research was supported by the Seoul R&BD program, Korea.

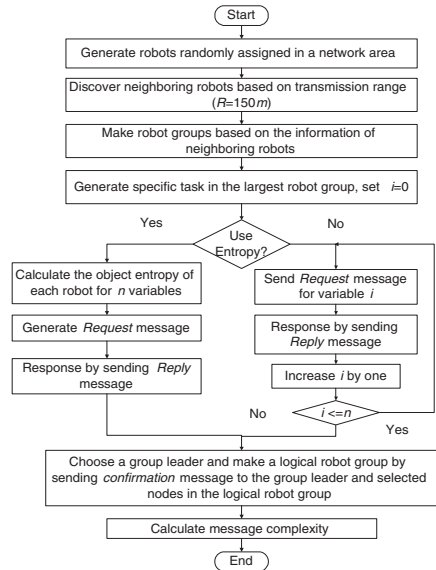
distributed and unconscious; (2) distributed and conscious; (3) intermediate and hierarchical; (4) centralized. The proposed multi-robot cooperative system uses a set of heterogeneous mobile robots, and the system employs a decentralized autonomous system. In this paper, an entropy concept from an information theory is used particularly for the cooperation of robots. The concept can be extended to relations between humans and robots in the society as well. For the past few years, MANETs have been emphasized as an emerging research area due to the growing demands for mobile and pervasive computing, where the dynamic topology for the rapid deployment of independent mobile users such as mobile robots becomes a new factor to be considered. One of the outstanding features of MANETs could be the self-creating, self-administrating, self-organizing, and self-configuring multihop wireless network characteristic. MANETs differ from conventional cellular networks because all links are wireless and the mobile users communicate with each other without using a base station. A MANET can be represented as an undirected graph  $G(\mathbf{V}, \mathbf{E})$  where  $\mathbf{V}$  is a finite nonempty set of nodes, which can be represented as  $\mathbf{V} = \{V_1^G, V_2^G, \dots, V_W^G\}$  where  $|\mathbf{V}| = W$  and  $\mathbf{E}$  is a collection of pairs of distinct nodes from  $\mathbf{V}$  that form a link, which can be represented as  $\mathbf{E} = \{E_1^G, E_2^G, \dots, E_W^G\}$ . A connected, acyclic, undirected graph which contains all nodes is defined as a free tree. Figure 1 (a) shows MANET nodes at an instantaneous time where it is composed of six partitioned subgraphs. One important issue in MANETs is the time-varying network topology which may be unpredictable over time, therefore, MANET routing algorithms keep updating neighbors discovery and inform nodes of the network topology change with node mobility. Based on the many considering factors of a MANET, the reduction of routing overhead is a main concern when a MANET routing protocol is developed. Therefore, one essential measure of the quality of a MANET routing protocol is the scalability in regards to an increase of the MANET nodes. Message complexity is defined as a performance measure where the overhead of an algorithm is measured in terms of the number of messages needed to satisfy the algorithm's request. In [4], Shen uses the message complexity to statistically measure the performance of the Cluster-based Topology Control (CLTC) protocol. The authors in [5] calculate the storage complexity and communication complexity to analyze the scalability of various MANET routing protocols and introduce the routing overhead of periodically updated LS messages, which follow the order of  $O(N^2)$ , where  $N$  indicates the number of nodes in a MANET.

## 2 Related Works

When a MANET is used as the communication method of mobile multi-robot system, the Fault-tolerant Configuration Algorithm [6] that makes the robots communicate one another is proposed. In addition, Time Complexity is used in order to measure the performance of a proposed algorithm. Winfield used MANET for the robot communication [7]. Robots distributed in a certain area use wireless transmitters and receivers to perform transmitting and receiving sensor data. In this case, various Ad hoc protocols are introduced for an efficient



(a) Mobile ad hoc network (MANET)



(b) Flowchart for developing a simulator

**Fig. 1.** MANET and Flowchart for developing a simulator

data transmission which requires less power than other protocols [8]. Mobile Robot Distance Vector (MRDV) and Mobile Robot Source Routing (MRSR) were developed for the efficient routing among Mobile Robot [9]. These protocols have less control overhead compared to the traditional Dynamic Source Routing (DSR) and Ad hoc On Demand Distance Routing (AODV) protocols. In order to improve the communication performance, an algorithm proposed to control the mobility of robot, where the scheduling is adopted in order to control robot mobility that causes blocking of communication [10]. Since MANET nodes can be replaced with mobile robots, in this paper, MANET grouping algorithms are applied to robot grouping in order to perform a specific task.

### 3 Proposed Algorithm

In this section, the mathematical background of robot grouping is introduced and a node is represented as a mobile robot and a route is represented as a robot group. The authors of [11] use the concept of entropy in order to measure the route stability of a MANET. In this paper, the route stability of a MANET is redefined to perform the robot grouping since a route is composed of several intermediate nodes between source and destination nodes and each node in a route has its own entropy value. In order to perform grouping algorithm and leader selection, two methods such as  $\gamma^j = GA_{k,l}^j(t, \Delta t)$  and  $\delta^j = LSA_{k,l}^j(t, \Delta t)$  can be defined as below [11]. The velocity vectors of node  $m$  and  $n$  are denoted by  $v(m, t)$  and  $v(n, t)$  respectively at time  $t$ . Therefore, the relative velocity  $v(m, n, t)$  between node  $m$  and  $n$  at time  $t$  is defined as [11],

$$v(m, n, t) = v(m, t) - v(n, t) \tag{1}$$

A variable  $a_{m,n}$  can be defined as below, where  $N$  is the discrete number in a time interval  $\Delta_t$ .

$$a_{m,n} = \frac{1}{N} \sum_{i=1}^N |v(m, n, t_i)| \tag{2}$$

Based on the relative velocity and the variable  $a_{m,n}$ , the entropy of node  $m$  of time interval  $\Delta_t$ , which is expressed as  $H_m(t, \Delta_t)$ , can be represented as [11],

$$H_m(t, \Delta_t) = \frac{-\sum_{k \in F_m} P_k(t, \Delta_t) \log P_k(t, \Delta_t)}{\log C(F_m)} \tag{3}$$

where  $P_k(t, \Delta_t) = a_{m,k} / \sum_{i \in F_m} a_{m,i}$ ,  $F_m$  is a set of neighboring nodes of a node  $m$  and  $C(F_m)$  is a degree of the set  $F_m$ . If the value of  $H_m(t, \Delta_t)$  is large, it means that route is stable, if the value of  $H_m(t, \Delta_t)$  is small, it means that route is unstable. In general, MANET consists of intermediate nodes between source and destination. Therefore, the grouping algorithm (GA) can be represented as  $\gamma^j = GA_{k,l}^j(t, \Delta_t)$  where  $\Sigma j = N_G$  and  $N_G$  is the number of robot grouping between two nodes  $k$  and  $l$ .

$$\gamma^j = GA_{k,l}^j(t, \Delta_t) = \prod_{i=1}^{N_r} [H_i(t, \Delta_t)] \Big|_j \tag{4}$$

where  $N_r$  indicates the number of intermediate nodes between two nodes of  $k$  and  $l$  of a single route (a single robot grouping)  $j$ . Based on  $\gamma^j = GA_{k,l}^j(t, \Delta_t)$ , the entropy of each intermediate node is multiplied and represented as  $\gamma^j$ . Then  $\gamma^j$  will be used to select which robot group has the best  $\gamma$  among several robot groups for a specific task. In a robot society, it can be assumed that there are several robot groupings for a specific work. In that case, after the  $\gamma$  value of each route is calculated, the route having the lowest  $\gamma$  value is selected and then the intermediate nodes of the route will perform the specific work.

$$\delta^j = LSA_{k,l}^j(t, \Delta_t) = \min_{i=[1,2,\dots,N_r]} [H_i(t, \Delta_t)] \Big|_j \tag{5}$$

The value of  $\delta^j$  can be used in selecting a group leader which has the lowest value of entropy among nodes in the selected route.

## 4 Experiment and Observation

In order to analyze the logical robot grouping and group leader selection proposed in this paper, a computer-based simulator has been developed. In addition, a standalone mobile robot environment is implemented, where the robots have no connection to an external network like the Internet. The discrete-event simulator was developed in Matlab in order to verify the various network topologies

and to calculate the message complexity. Figure 1(b) shows the flowchart of the proposed algorithm implemented in the simulator. In the computer simulator, robots are randomly distributed with uniform density in a network area of  $1km^2$ . The transmission range of a robot is defined as  $150m$ . The simulator is composed of several modules as shown in Fig. 1(b). In this paper, variable status of a robot, which represents its own power, mobile speed, the capability for a certain task is represented as a single entropy value, which is called as an object entropy. The equation (3) represents the object entropy composed by the speed variable only. First, a random node generator in the simulator generates random robots. Each robot detects its neighboring robot based on the transmission range, which finally consists of a communication robot group ( $C_R$ ). After the group is constructed, only a single robot, which needs to get cooperation from other robots, is selected to trigger a specific work and transmits a *request* message into the group. Other robots in the group check the required entropy value, which is a threshold, from the source robot. Only the robots whose current entropy values are below the threshold value send a *reply* message back to the source robot. Robots responding by sending the *reply* message consist of a logical robot grouping within the communication robot group and they help the source robot in order to perform the specific task. All the robots generating a *reply* message consist of a logical robot group ( $L_R$ ). Therefore, between the communication robot group and the logical robot group, the relation of  $n(L_R) \leq n(C_R)$  is constructed where  $n(A)$  represents the number of element in a set  $A$ . In the case of not using the entropy in Fig. 1(b), it is assumed that  $n$  variables need to be checked in order to perform a specific task and select a group leader in the logical robot group. A *request* message for each single variable  $i$  is generated and sent to all robots. Whenever a source robot issues a *request* message, all the robots receiving the *request* message send a *reply* message back to the initiator. Therefore, the source robot can compare the status of all robots for the specific variable, such as a remained power or mobile speed. Based on the  $n$  iterations of the above procedure such as sending *request* messages and receiving *reply* messages, the source robot can determine robots that consist a logical robot group destined to perform a required specific task. After deciding the members, the source robot sends a *confirmation* message to the member robots selected in the logical robot group by unicasting pattern. The *confirmation* message includes a group leader robot that controls the entire logical robot group in charge of the source robot since the selected robot leader has the capacity to satisfy the required  $n$  variables. In the case of using entropy, since all nodes represent its status as a single variable that is object entropy, there is no need for the source robot to repeat the  $n$  times of iteration procedure in order to make the logical robot grouping. So to speak, the  $n$  variables are represented as a single entropy value in this case. Therefore, one time of sending a *request* message and receiving *reply* messages is enough for the initiator to check the status of all robots in order to perform the specific task. Same as the no-entropy case, after deciding the members, the source robot sends a *confirmation* message to the member robots selected in the logical robot group by unicasting pattern. In



the simulation, it is assumed that only a single variable is included in a *request* message in order to consider the maximum length of the *request* message. It is assumed that entropy of each robot is randomly assigned and the value is kept as a constant when each robot performs logical robot grouping and selecting a group leader. In addition, the message will not be lost in the middle of transmitting at the intermediate nodes. In this simulation, the number of message that is used for the communication robot grouping is not counted as the performance metric of network resource. However, the number of message that is used for the logical robot grouping and selecting a group leader is counted as the performance metric of network resource and depicted in following result graphs. Based on a random network topology scenario, the number of nodes and the network topology are not changed during the simulation in order to analyze the message complexity of each variable case and compare the simulation results. For simplicity, 5 variables, which means that  $n$  equals 5, are selected in this simulation. Figure 2(a) shows the results of message complexity of 5 different variables respectively. *Monte Carlo* simulation performing 400 iterations at each variable is used to evaluate various network topologies for random scenarios and to calculate the average number of messages for each variable where each topology is composed of a different number of free trees. The random robot generator used in this simulation and simulator performance was verified for the numbers of robots 100, 125, 150, and 175 so that the average number of robots per cluster as well as several specs in the adaptive dynamic backbone (ADB) algorithm [4] matched with the results in [4], which was performed by QualNet, with less than a 1% difference for almost all cases [12]. In Fig. 2 (a),  $x$  axis shows the number of robots in a communication robot grouping and  $y$  axis shows the number of message so as to perform a logical robot grouping and to select a group leader. The graph marked as *Use Entropy* indicates the scenario when all robots are prepared with entropy when a source robot generates a *request* message. The graphs marked as *No Use Entropy* ( $n$  variables) indicates the scenarios when robots do not use the concept of entropy. Therefore, in order to make a logical robot grouping and to select a group leader, a source robot needs to send a *request* message for a specific requirement that is called as a variable. Since  $n$  variables are needed to perform the logical robot grouping, the robot repeatedly sends *request* message  $n$  times where each *request* message includes a different variable. Whenever, robots receiving the *request* message, they should response by sending a *reply* message to the initiator robot in order to report their current status asked by the *request* message for a specific variable. As it can be expected and shown in Fig. 2(a), the message complexity of the scenario using entropy is much less than the message complexities of the other scenarios using no entropy. In addition, it can be shown that if the number of the required variable is increased in the scenarios of not using entropy, the number of message used is also increased. Therefore, based on the above results, it can be concluded that with the use of entropy, robots can make an logical robot grouping more efficiently since the network resource such as the number of message using the contention-based wireless communication channel can be significantly reduced

and the power used for sending and receiving messages is also saved. During the simulation, each topology is composed of a different number of communication robot groups. Even though several communication robot groups based on the transmission range of robot exist in a network topology, only the largest communication robot group is considered in the simulation. Therefore, from the graph of the *Use Entropy*, it is shown that the number of message is not so high when the number of the robots is composed of 10. In the case of using the entropy,  $n$  numbers of variables are represented by single object entropy. Since  $n$  variables need to be converted as object entropy, the operational complexity of the CPU to calculate the object entropy is a little higher than the one that does not use the object entropy. However, the operational complexity of the CPU can be ignored due to the improvement of CPU capability and the reduction of CPU production cost. Therefore, this paper does not consider the CPU operational complexity when the object entropy is calculated in each robot. Figure 2 (b) compares the percentage overhead of the message complexity between *Use Entropy* and *No Use Entropy*. In the case of *No Use Entropy*, 2 variables, 3 variables, 4 variables, and 5 variables have the 100%, 200%, 300%, and 400% more overhead compared to the *No Use Entropy*. In addition, it is shown that the case of *No Use Entropy*(1 variable) is same as the case of *Use Entropy*.

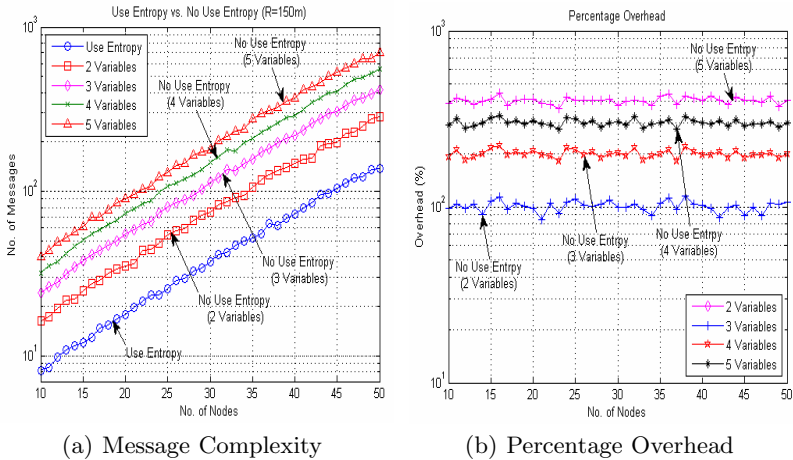


Fig. 2. The comparison of Message Complexity & Percentage Overhead

## 5 Conclusion

This paper proposes a novel method in order to make a logical robot grouping and to select a group leader based on the equations (4) and (5) using the concept of object entropy and MANET in order to perform a specific task. The instantaneous entropy value of robot is used as the object entropy in the simulation. Since the object entropy is used as a representative value for the current robot

status, it uses the network resources efficiently, which uses the contention-based wireless communication channel. In addition, it can save the power of mobile robots since robots can reduce the number of communication. This paper also shows the process of making the logical robot group and selecting a group leader. When there are several routes between source and destination robots, a route having the lowest value of  $\gamma^j = GA_{k,l}^j(t, \Delta_t)$  is selected. Moreover, it is shown that based on the value of  $\delta^j = LSA_{k,l}^j(t, \Delta_t)$ , a group leader is selected.

## References

1. F.R. Noreils: Toward a robot architecture integrating cooperation between mobile robots: Application to indoor environment. *International Journal of Robotics Research*, vol. 12. (1993) 79–98.
2. R.R. Murphy, C.L. Kisetti, R. Tardif, L. Irish, A. Gage: Emotion-based control of cooperating heterogeneous mobile robots. *IEEE Trans. Robotics and Automation*, vol. 18, (2002) 744–757.
3. A. Borkowski, M. Gnatowski, J. Malec: Mobile robot cooperation in simple environments. *Second Workshop on Robot Motion and Control*, (2001) 109–114.
4. C.-C. Shen, C. Srisathapornphat, R. L. Z. Huang, C. Jaikaeo, E. L. Lloyd: CLTC: A cluster-based topology control framework for ad hoc networks, *IEEE Trans. Mobile Computing*, vol. 3, no.1, (2004) 18–32.
5. X. Hong, K. Xu, M. Gerla: Scalable Routing Protocol for Mobile Ad Hoc Networks, *IEEE Network*, (2002) 11–21.
6. P. Basu, J. Redi: Movement Control Algorithms for Realization of Fault-tolerant Ad Hoc Robot Networks, *IEEE Network*, (2004) 36–44.
7. A. F. 1. Winfield: Distributed Sensing and Data Collection via Broken Ad Hoc Wireless Connected Networks of Mobile Robots, *Distrib. Autonomous Robotic SFS*. 4, L. E. Parker, G. Bekey, and I. Borhen, Edr., Springer. (2000) 273–282.
8. Z. Wang, M. Zhou, N. Ansari: Ad-hoc Robot Wireless Communications, *IEEE Conference on System, Man and Cybernetics*, vol. 4, (2003) 4045–4050.
9. S. M. Das, Y.C. Hu, C.S.Lee, Y.H.Lu: Efficient Unicast Messages for Mobile Robots, *IEEE Conference on Robotics and Automation*, (2005) 93–98.
10. P. Bracka, S. Midonnet, G. Roussl: Trajectory based communication in an ad hoc network of robots, *IEEE Conference on Wireless and Mobile Computing, Networking and Communication*, vol. 3, (2005) 1–8.
11. B. An, S. Papavassiliou: An Entropy-Based Model for supporting and Evaluating Route Stability in Mobile Ad hoc Wireless Networks, *IEEE Comm. Letters*, vol.6, no.8, (2002) 328–330.
12. S.-C. Kim and J.-M. Chung: Message Complexity Analysis of Mobile Ad Hoc Network Address Autoconfiguration Protocols, *IEEE Trans. Mobile Computing*, submitted for the second review of publication.

# Robot Path Planning in Kernel Space

José Alí Moreno and Cristina García

Universidad Central de Venezuela, Laboratorio de Computación Emergente,  
Facultades de Ciencias e Ingeniería, Venezuela  
{jose, cgarcia}@neurona.ciens.ucv.ve

**Abstract.** We present a new approach to path planning based on the properties of the minimum enclosing ball (MEB) in a reproducing kernel space. The algorithm is designed to find paths in high-dimensional continuous spaces and can be applied to robots with many degrees of freedom in static as well as dynamic environments. In the proposed method a sample of points from free space is enclosed in a kernel space MEB. In this way the interior of the MEB becomes a representation of free space in kernel space. If both start and goal positions are interior points in the MEB a collision-free path is given by the line, contained in the MEB, connecting them. The points in work space that satisfy the implicit conditions for that line in kernel space define the desired path. The proposed algorithm was experimentally tested on a workspace cluttered with random and non random distributed obstacles. With very little computational effort, in all cases, a satisfactory free collision path could be calculated.

## 1 Introduction

Work in path planning has not only impacted robotics, but also arise in a large number of fields including operations research, scheduling, graphics animation, surgical planning or computational biology, see [1] for a review. In any case, research in robot motion planning remains as one of the important fields of study in the task of building autonomous or semi-autonomous robot systems. The path planning problem involves computing a continuous sequence (“a path”) of configurations (generalized coordinates) between an initial configuration (start) and a final configuration (goal) while respecting certain constraints imposed by a complicated obstacle distribution. This definition of the problem simplifies some of the aspects of robot motion planning. The dynamic properties of the robot can be ignored and the problem is transformed to a purely geometrical path planning problem. The formal definitions of the concepts involved in path planning can be seen in the books [2], [3].

In this work we will concentrate on the most basic version of the path planning problem which involves computing a collision-free path between two configurations in a static environment of known obstacles. This single query path planning problem has received considerable attention from the robotic community. In consequence a broad class of complete and incomplete algorithms designed

over different technologies and general approaches have resulted [2], [3], [4], [5], [6], [7], [8], [9], [10]. It is argued [11] that the complete path planning problems are mostly NP-hard. Generally speaking, the complexity of the problem is exponential in the number of degrees of freedom (DOF) of the robot, and polynomial in the number of obstacles in the environment. This result implies that complete algorithms are only practical for problems with a low dimensional configuration space. The complexity of complete path planning methods lead researchers to seek heuristic methods with weaker notions of completeness, such as probabilistic completeness. The accepted tradeoff is that the methods are incomplete, but will find a solution with any probability given sufficient running time, the need is then for methods that converge rapidly. The algorithm proposed in the present work deals with this complexity issue in a different way. First of all the algorithm is deterministic, finds an approximate representation of free space ( $C_{\text{free}}$ ) in a generalized space and calculates there the solution, all this with a small time complexity. A solution is always calculated whenever it exists.

The algorithm is based on the “kernel trick”, an idea that has been extensively used in machine learning to generate non-linear versions of conventional linear algorithms [12], [13]. The procedure works as follows: if the input points enter in the algorithm to be generalized only in the form of dot products  $\langle \mathbf{x} \cdot \mathbf{y} \rangle$ , then they can be replaced by a kernel function  $K(\mathbf{x}, \mathbf{y})$ . A symmetric function  $K(\mathbf{x}, \mathbf{y})$  is a kernel if it fulfills Mercer’s condition, i.e. the function  $K$  is (semi) positive definite. When this is the case there exists a mapping  $\Phi$  such that it is possible to write  $K(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}) \rangle$ . The kernel represents a dot product on a different space  $\mathbb{K}$  called kernel space into which the original points are mapped. That is, a kernel function defines an implicit mapping of input points into a high or infinite dimensional kernel space and allows the algorithm to be carried out there. In the present work the “kernel trick” applied to the minimum enclosing ball (MEB) algorithm in [14] is used to produce an approximate representation of free space in  $\mathbb{K}$ . This is achieved by enclosing a representative sample of free space points by a kernel MEB defined by a subset of the sampled points. Since the interior points of the MEB can be assumed to belong to free space a collision-free path is given by the line, inside the ball, connecting the start and goal positions. The points that satisfy the implicit conditions for that line in kernel space define the path in workspace.

The organization of the paper is as follows: In the next section the MEB algorithm is introduced. In Sect. 3 the generalization of the minimum enclosing ball algorithm with the “kernel trick” is described. In Sect. 4 the proposed algorithm is presented and its initialization and parameterization is explained. In Sect. 5 experimental results are presented. Section 6 is for the conclusions.

## 2 Minimum Enclosing Ball

In its most general form, the MEB problem deals with the computation of a minimum radius ball that encloses a given set of objects (points, balls, etc.) in  $\mathbb{R}^d$ . In this paper we limit ourselves to the case of enclosing a set of points.

Following [15], let  $B_{c,r}$  denote a ball of radius  $r$  centered at point  $c \in \mathbb{R}^d$ . Given an input set  $S = \{p_1, \dots, p_n\}$  of  $n$  points in  $\mathbb{R}^d$  the minimum enclosing ball  $MEB(S)$  of set  $S$  is the unique minimum radius ball containing  $S$ . It is worth noticing that although the MEB is unique the set of points defining the MEB not necessarily is. The center  $c^*$  of the  $MEB(S)$  is called the *1-center* of  $S$ , since it is the point of  $\mathbb{R}^d$  that minimizes the maximum distance to points in  $S$ .

Instead of calculating an exact MEB solution, in general for practical purposes an approximated solution is enough. This approach is taken in [14]: Let  $r^*$  denote the radius of the  $MEB(S)$ . A ball  $B_{c,r}$  containing  $S$  is said to be a  $(1 + \epsilon)$ -approximation of the  $MEB(S)$  if  $r \leq (1 + \epsilon)r^*$  and  $\epsilon > 0$ . Clearly,  $\epsilon$  is the parameter that specifies the tolerance of the solution.

In [16] the core-set concept is introduced: Given a subset,  $X \subseteq S$ , and a value  $\epsilon > 0$ , a *core-set* of  $S$  has the property that the smallest ball containing  $X$  is within  $\epsilon$  of the smallest ball containing  $S$ . That is, if the radius of the smallest ball containing  $X$  is expanded by  $1 + \epsilon$  then the expanded ball contains  $S$ :  $B_{c,(1+\epsilon)r} \supset S$  where  $B_{c,r} = MEB(X)$ , thus the ball  $B_{c,(1+\epsilon)r}$  is a  $(1 + \epsilon)$ -approximation of  $MEB(S)$ . In [14] is demonstrated that a bound for the core set size is  $1/\epsilon$  and that it is a tight bound in the worst case. They also present a simple algorithm for computing the 1-center. Let  $r_i \equiv r_{B(S_i)}$  and  $c_i \equiv c_{B(S_i)}$ , select an arbitrary point  $p_{f0} \in S$  and set:  $c_1 = p_{f0}$  and  $X_1 = \{c_1\}$ . Repeat the following step  $1/\epsilon^2$  times: find the point  $p_{fi} \in S$  farthest away from  $c_i$  and move toward  $p_{fi}$  by:  $c_{i+1} \leftarrow c_i + (p_{fi} - c_i) \frac{1}{i+1}$  adding it to the actual working set:  $X_{i+1} \leftarrow X_i \cup \{p_{fi}\}$ . This procedure has a time complexity of  $O(dn/\epsilon^2)$ .

We propose to stop the algorithm when the farthest point found is within the specified  $\epsilon$  tolerance. Under such circumstances, the center  $c_k$  can be expressed as a combination of the points being included step by step. In fact, the center  $c_k$  corresponds to the centroid of the points in the actual working set:

$$c_k = \frac{1}{k} \sum_{j=0}^{k-1} p_{fj} \tag{1}$$

Since a new point is inserted to the working set on every iteration, the size of the core-set (working set when algorithm stops) equals the number of iterations. It can be noted that the center of the MEB is a convex combination of the points in the core-set. The experiments show that the resulting core-sets are smaller than the bound of  $1/\epsilon$  [15].

This MEB algorithm is a straightforward procedure for the space where the sample points are embedded. To use it for solving MEB in an alternate high dimensional feature space we make use of kernel functions, as applied in learning machine problems.

### 3 MEB in Kernel Space

A non trivial representation of free space can be obtained by transforming the points in  $C_{free}$  to a high dimensional space and calculating the minimum enclosing ball of those points there. In this way the points in the interior of the MEB

would be the desired representation. The main concern to achieve this is to find a nonlinear transformation to an *appropriate* high dimensional space.

Lets suppose that a suitable transformation is known. Given  $S$ , a set of  $n$  sample points of  $C_{\text{free}}$  in a general  $d$ -dimensional configuration space, a suitable nonlinear transformation  $\Phi$  from  $\mathbb{R}^d$  to some high dimensional kernel space is applied. In the kernel space, a ball with center  $\mathbf{c}$  and radius  $r$  that encloses all the points, can be written in terms of the Euclidean norm as:

$$\|\Phi(x_i) - \mathbf{c}\|^2 \leq r^2 \quad \forall i = 1, \dots, n . \tag{2}$$

Thus to solve the MEB problem in the kernel space it is necessary to find the center  $\mathbf{c}$  and the radius that minimizes (2). If the set of transformed points:  $S_{tr} = \{\Phi(p_1), \dots, \Phi(p_n)\}$  is known, this can be accomplished using the simple algorithm for 1-center described above.

### Kernel Trick

The distance of a point to the center of the ball in kernel space is expressed using dot products:

$$R^2(x) = \|\Phi(x) - \mathbf{c}\|^2 = \langle (\Phi(x) - \mathbf{c}) \cdot (\Phi(x) - \mathbf{c}) \rangle . \tag{3}$$

Instead of calculating the dot product in kernel space, a kernel function can be introduced such that:

$$K(x_i, x_j) = \langle \Phi(x_i) \cdot \Phi(x_j) \rangle . \tag{4}$$

Not all functions are kernel functions in the sense used here, for a comprehensive treatment of kernel functions the reader is referred to [12] and references therein. For the sake of simplicity, we just say that a positive symmetric function  $K(x_i, x_j)$  that satisfies Mercer’s theorem, represents dot products in a feature space.

The ‘kernel trick’ consists then in the substitution of a dot product by a kernel function. It can be applied in any relation expressed in terms of dot products [12].

The application of the kernel trick to the MEB algorithm is straightforward. Let the center of the ball in kernel space be expressed by:

$$\mathbf{c} = \alpha \sum_i \Phi(p_i) \quad \forall i = 1, \dots, m , \tag{5}$$

with points  $\Phi(p_i)$  in the core-set of cardinality  $m$  and  $\alpha = 1/m$ . By replacing (5) into (3) and using kernels instead of dot products we obtain:

$$R^2(x) = K(x, x) - 2\alpha \sum_i K(p_i, x) + \alpha^2 \sum_i \sum_j K(p_i, p_j) . \tag{6}$$

Hence, at each step, the farthest point is the one that maximizes (6). Thanks to the kernel functions the search is made directly in input space. It can be seen that the worst case complexity of the resulting algorithm is  $O(dn/\epsilon^2)$ , where  $d$  stands for the dimension of configuration space.

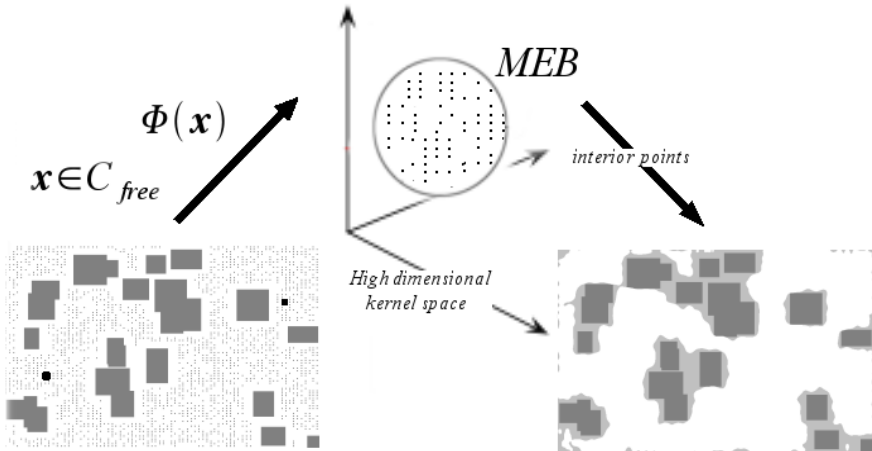
For the application of this approach it is important to select an adequate kernel function. In this work it was experimentally found that the inverse multiquadric (7) kernel is an appropriate selection. The,  $\sigma$  denotes a width parameter that must be fixed a priori.

$$K(x_i, x_j) = \frac{1}{\sqrt{\frac{\|x_i - x_j\|^2}{\sigma^2} + 1}} \quad (7)$$

### 4 The Algorithm

The algorithm consists on a very simple procedure. Initially a set of  $n$  points from free space  $C_{free}$  is selected by simple random sampling. A suitable kernel function and its associated width parameter  $\sigma$  are chosen. Finally the minimum enclosing ball for the set of sample points is calculated in kernel space by the procedure explained in Sect. 2 and Sect. 3. The points in the volume of this MEB conform an approximate representation of  $C_{free}$  in kernel space. If both, the start and goal positions, belong to the interior of the MEB, the collision-free path is given by the line, contained in the MEB, connecting the start and goal positions. The points in work space that satisfy the implicit conditions for that line in kernel space define the desired path. If one or both, start and/or goal positions, are not in the interior of the MEB there is no such collision-free path.

The initialization and parameterization of the algorithm is straightforward once the kernel function is selected, which implies the specification of the kernel parameter  $\sigma$ . The algorithm involves only two additional parameters: The number of randomly sampled points of  $C_{free}$  and the tolerance for calculating the MEB. The values used in all the experiments were:



**Fig. 1.** Principal steps of the algorithm. (Left) workspace, (center) kernel space and MEB (right) approximate representation of  $C_{free}$ .

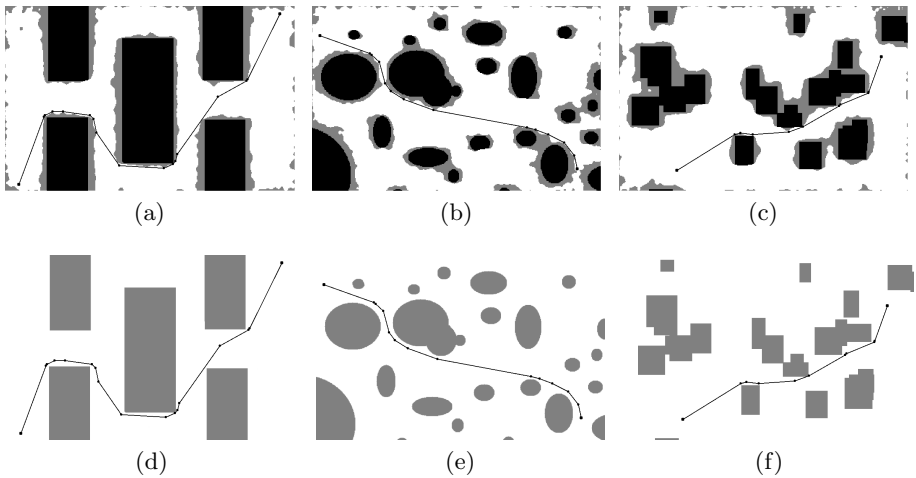


- kernel parameter  $\sigma$  between 0.002 and 0.008
- number of sampling points:  $\sim 4000$  for 2 DOF and  $\sim 40000$  for 3 DOF.
- tolerance  $\epsilon$  of the MEB:  $1 \times 10^{-6}$

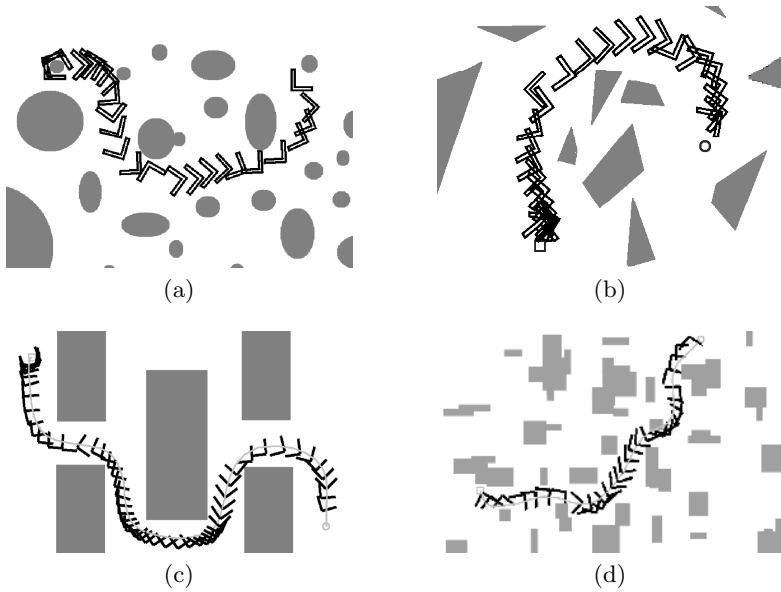
The important stages of the algorithm for a point robot in a 2D workspace are depicted in Fig. 1. The left image shows the original workspace with the sample points from  $C_{\text{free}}$  and the start and goal positions. The image in the center represents the implicit mapping of the free-space sampled points to the MEB in kernel space. The image to the right shows the approximate representation of  $C_{\text{free}}$  obtained from the points in the interior of the MEB (white region). The gray regions are points not belonging to the interior of the MEB.

## 5 Experiments

The performance of motion planning algorithms is generally described and evaluated with a set of test problems. Currently, there exists no commonly used standard test tasks for evaluating motion planning methods. Typically, the performance of a motion planning algorithm is described by reporting its running time on a set of tasks chosen by the developer as being suitable for demonstration. In this sense, in order to evaluate the algorithm's path generation capability a number of simulations, for 2 and 3 DOF robots on several 2D workspaces with different obstacle distributions, have been conducted. The algorithm has been implemented in C++ and compiled with GNU C++ on a 2.4 GHz Pentium IV based machine running Linux. The images used to depict the 2D workspace had a size of  $800 \times 500$  pixels and in the case of the 3 DOF robots for the rotation



**Fig. 2.** Several path planning examples in different scenarios: (a),(b),(c) calculated path in the approximate  $C_{\text{free}}$  (d),(e),(f) same solutions depicted in actual workspace



**Fig. 3.** Several path planning examples for a 3 DOF robot in different scenarios

angle in  $[0, 2\pi]$  a discretization of 1.8 degree is used. In all cases for the calculation of the MEB representation of  $C_{\text{free}}$  of the 2 DOF robots a set of  $\sim 4000$  and for the 3 DOF robots  $\sim 40000$  randomly selected sample points are used. This means that the path generation requires in general the sampling of a small fraction of the points in configuration space.

Figure 2 depicts the resulting collision free paths for a point robot on several 2D workspaces cluttered with obstacles. It can be seen that the algorithm produces, in all cases, reasonable good path solutions. The average execution time for these experiments, over 20 runs, is 300 msec. This is to be compared with the average 350 ms. obtained on the same workspaces with the rapidly-exploring random tree method (RRT) in [10].

Figure 3 shows the path planning results for an L shaped 3 DOF robot on several 2D workspaces. As before in all cases the algorithm generates good collision free paths in average execution times, over 20 runs, of 5 sec. The increased execution times observed in these experiments are a direct consequence of the larger set of sampled points used in the MEB representation of  $C_{\text{free}}$ .

## 6 Conclusion

A new robot path planning algorithm based on the properties of the minimum enclosing ball in a reproducing kernel space is proposed. It is designed to find paths in high-dimensional continuous spaces and can be applied to robots with

many degrees of freedom in static as well as dynamic environments. In the procedure a generalized MEB algorithm is applied to a small set of sample points of free space. Then the points in the interior of this MEB conform a representation of  $C_{\text{free}}$  in kernel space. If both, the start and goal positions, belong to the interior of the MEB, the collision-free path is given by the line, contained in the MEB, connecting the start and goal positions. The points in work space that satisfy the implicit conditions for that line in kernel space define the desired path. The algorithm is computationally very simple and computer experiments show that the method yields very efficient performance on simulated scenarios. It does not require the application of complex procedures like special search methods, collision checking procedures, the optimization of global functions or any tessellation of the workspace. The algorithm was experimentally tested for a nonholonomic 2 DOF robot and free flying L 3 DOF robots on different random generated workspace configurations. In all cases the observed performances were very good. The algorithm can be straightforwardly applied to more complicated situations like robots with higher degrees of freedom or articulated robotic arms in 3 dimensional workspaces. Although we consider that the proposed algorithm has proven to be very successful in the preliminary experiments presented, there are some issues pending that could further improve its performance. Experimental evaluation of these issues constitute the basis of future research:

- The size of the obstacles can be artificially and conveniently grown in order that the planned paths do not pass too near the obstacles. In this way uncertainties of the robot and obstacles positions could be accounted for.
- Alternative sampling strategies must be considered in order to reduce the time complexity of the algorithm without losing information of the possible narrow pathways.
- Further reduction in the computation time could be achieved by introducing efficient farthest point algorithms in the calculation of the MEB.
- Experiments extending the approach to higher DOF systems should be carried out.

## References

1. Latombe, J.: Motion planning: A journey of robots, molecules, digital actors and other artifacts. *Journal of Robotics Research*, Especial Issue on Robotics at the Millenium **18**(Part II) (1999) 1119–1128
2. Latombe, J.: *Robot Motion Planning*. Kluwer Academic Publisher, Boston, Mass (1991)
3. LaValle, S.: *Planning algorithms*. available at <http://misl.cs.uiuc.edu/planning> (2004)
4. Caselli, S., Reggiani, M., Rocchi, R.: Heuristic methods for randomized path planning in potential fields. *IEEE International Symposium on Computational Intelligence in Robotics and Automation* (2001) 426–431
5. Amato, N., Wu, Y.: A randomized roadmap for path manipulation planning. *IEEE International Conference on Robotics and Automation* (1996) 113–120

6. Kavraki, L., Latombe, J.: Randomized preprocessing of configurations space for path planning. *IEEE International Conference on Robotics and Automation* (1994) 2138–2139
7. Behring, C., Bracho, M., Castro, M., Moreno, J.: An algorithm for robot path planning with cellular automata. In: *Theoretical and Practical Issues on Cellular Automata*. Springer-Verlag, Berlin (2000) 11–19
8. Bracho, M., Moreno, J.: Heuristic algorithm for robot path planning based on real space renormalization. *Lecture Notes in Artificial Intelligence* **1952** (2000) 379–388
9. Moreno, J., Castro, M.: Heuristic algorithm for robot path planning based on a growing elastic net. *Lecture Notes in Computer Science* **3808** (2005) 447–454
10. LaValle, S.: Rapidly-exploring random trees: A new tool for path planning. Technical Report Technical Report TR 98-11, Computer Science Dept. Iowa State Univ. (Oct. 1998)
11. Canny, J.: *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA (1988)
12. Scholkopf, B., Smola, A.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. The MIT Press, Cambridge, Mass (2002)
13. Scholkopf, B., Burges, J., Smola, A.: *Advances in Kernel Methods - Support Vector Learning*. The MIT Press, Cambridge, Mass (1999)
14. Badoiu, M., Clarkson, K.L.: Optimal core-sets for balls. In: *DIMACS Workshop on Computational Geometry*. (2002)
15. Kumar, P., Mitchell, J., Yildirim, E.A.: Computing core-sets and approximate smallest enclosing hyperspheres in high dimensions. In: *Proceedings of the 5th Workshop on Algorithm Engineering and Experiments - ALENEX'03*. (2003)
16. Badoiu, M., Har-Peled, S., Indyk, P.: Approximate clustering via core-sets. In: *Proceedings of the 34th Symposium on Theory of Computing*. (2002)

# A Path Finding Via VRML and VISION Overlay for Autonomous Robot

Kil To Chong<sup>1</sup>, Eun-Ho Son<sup>1</sup>, Jong-Ho Park<sup>1</sup>, and Young-Chul Kim<sup>2</sup>

<sup>1</sup>Division of Electronics and Information Engineering, Chonbuk National University,  
Duckjin-Dong, Duckjin-Gu, Jeonju 561-756, Korea  
{kitchong, yauchi1, jhpark}@chonbuk.ac.kr

<sup>2</sup>Department of Mechanical Engineering, Kunsan National University, Korea  
kimyc@kunsan.ac.kr

**Abstract.** We describe a method for localizing a mobile robot in its working environment using a vision system and Virtual Reality Modeling Language (VRML). The robot identifies the landmarks located in the environment, using image processing and neural network pattern matching techniques, and then it performs self-positioning based on vision information and a well-known localization algorithm. The correction of position error is performed using the 2-D scene of the vision and the overlay with the VRML scene. Through an experiment, the self-positioning algorithm has been implemented to a prototype robot and also it performed autonomous path tracking.

## 1 Introduction

We address a self-localization problem for mobile robots operating in unknown or partially known environment. Obstacle avoidance is one of the key issues in applications of mobile robot system. To determine the path of a mobile robot it is necessary to know the position of robot, which can be determined using landmark-localization techniques. Inaccurate localization may lead a robot to dangerous conditions. Landmarks are any detectable structure in the physical environment [1]. In this paper, we used specially designed marks as landmarks. Given a specific focal length and a single image of three landmarks, it is possible to compute the angular separation between the lines of sight of the landmarks. Then, if the global positions of the landmarks are known, the angular separations can be used to compute the position of robot and heading relative to a 2-D floor map [3–5]. The simplicity of this approach, and the fact that it does not involve any 3-D reconstruction, has made it popular [1].

We applied several image-processing techniques to extract landmarks from the visual scene, and a neural network has been used for pattern recognition of landmarks. Then, a construction of 3-D/visual scene was modeled using Virtual Reality Modeling Language (VRML), a type of Web3D technology that supports 3-D information on the web. By overlapping the visual and 3-D scenes in wire-frame mode, the accuracy of self-positioning can be verified. Moreover, the invisible sides of obstacles (e.g., backside or inside) are predicted, thereby extending the field-of-vision. Invisible sides

can be predicted from the 3-D model. Furthermore, VRML can serve as a reference data source of the original environment information. Any displaced objects can be detected easily by comparing both images, so both the position and path of a robot can be detected in the 2-D scene. The proposed path-finding scheme is realized while avoiding the obstacles [14].

## 2 Landmark Pattern Recognition

Landmarks are any detectable structure located in the physical environment. The vertical lines are usually used for recognition, others may use specially designed markers, e.g., crosses or patterns of concentric circles. In this paper, specially designed markers are used as landmarks. They composed with simple patterns with colored background (Fig.1).

Image processing for recognizing landmarks are accomplished by the following steps.

1. Acquire image data from vision ( $320 \times 240$ , 24bit)
2. Apply color histogram stretching to the image for improvement.
3. Binary image processing.
4. Remove salt and pepper noise.
5. Detect objects using tracking algorithm.
6. Separate landmarks detected from original image.
7. Extract and resize central marker for pattern recognition.



**Fig. 1.** Landmarks

We applied a tracking algorithm for image processing appeared [7]. This algorithm is used to locate the peripheral boundary of the region. First, a raster scan searches the image until a specific color is found. If a scanned point has already been visited once, the next point is scanned. If the value of the point is a specific color, a peripheral boundary-tracking process is initiated in a clockwise direction. In this way, the entire region of a landmark in the image is determined. For pattern recognition, a central mark is extracted from the original image in accordance with the tracked region.

### 2.1 Pattern Recognition

In this section, defining features and neural networks structure for pattern recognition are described.

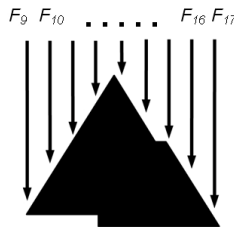
**Features of the Pattern**

For recognition of patterns using neural networks, the features of mark should be defined. We selected 17 features as inputs of the neural network. The first eight features are defined by the number of pixels in the eight directions ( $\rightarrow \searrow \downarrow \swarrow \leftarrow \nearrow \uparrow \nearrow$ ). The example is shown in Table 1.

**Table 1.** The example of defining features

	feature	# of pixel
	$F_1 (\rightarrow)$	$b$
	$F_2 (\searrow)$	$a+c$
	$F_3 (\downarrow)$	$0$
	$F_4 (\swarrow)$	$0$
	$F_5 (\leftarrow)$	$d+f$
	$F_6 (\nearrow)$	$0$
	$F_7 (\uparrow)$	$e$
	$F_8 (\nearrow)$	$g$

However, eight features are not sufficient to distinguish all patterns. For instance, it is difficult to distinguish a triangle ( $\blacktriangle$ ) from an inverted triangle ( $\blacktriangledown$ ) since both of them has the same eight features. Therefore, it is necessary defining other features to overcome this deficiency. The additional features are defined as shown in Fig. 2.



**Fig. 2.** Additional features

Each feature is defined by the number of pixels from the top to a mark which is a nonzero value pixel. Landmark shown in Fig. 2 can be defined by the following 17 features:

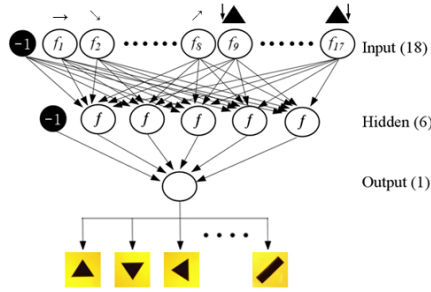
$$feature = [b, a + c, 0, 0, d + f, 0, e, F_9, F_{10}, \dots, F_{16}, F_{17}] \tag{1}$$

Each feature was normalized between 0 and 1.

**Neural Networks**

A back-propagation algorithm was used to train the MLP for pattern recognition. The neural network consists of three layers as shown in Fig. 3: there are 18 nodes in input layer, 6 neurons in hidden layer, and one node in the output layer. The neural net was trained with 50 different patterns for each landmark, and the activation function was.

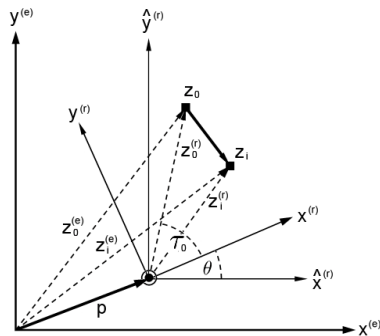
$$f(x) = \frac{1}{1 + \exp(x)} \tag{2}$$



**Fig. 3.** Structure of neural network

**2.2 Localization Algorithm**

We applied linear position estimation method for the self-positioning since the existing triangulation algorithms take too long for real-time navigation [4]. The key of linear method is obtaining a set of equations whose solution is a set of position estimates. The triangulation methods may also provide a set of position estimates, but the equations are nonlinear [4]. A landmark  $z$  shown in Fig. 4 can be written as a complex number in the robot-centered coordinate system. For each landmark  $z_i^{(r)}$  we can have an expression



**Fig. 4.** External coordinate



$$z_i^{(r)} = l_i e^{j\tau_i} \text{ for } i=1, \dots, n \tag{3}$$

where length  $l_i$  is the unknown distance of the robot to landmark  $z_i^{(r)}$ , angle  $\tau_i$  is the measured angle between  $z_i^{(r)}$  and the coordinate axis  $x^{(r)}$ , and  $j = \sqrt{-1}$ .

Note that using the robot-centered coordinate system that is spanned by axes  $\hat{x}^{(r)}$  and  $\hat{y}^{(r)}$  does not imply that the robot knows its orientation in the environment a priori. It does not know its orientation angle  $\theta = \angle(x^{(r)}, x^{(e)}) = \angle(x^{(r)}, \hat{x}^{(r)})$ . Instead, we can compute vector  $\hat{z}_o^{(r)}$  which can then be used to compute the orientation of the robot.

$$\theta = \angle(\hat{z}_o^{(r)}, \hat{x}^{(r)}) - \tau_0 \tag{4}$$

### 2.3 Application

We used Microsoft Visual C++6.0 for simulations, and Cortona Player 4.2 (Parallel Graphics) as the VRML client. Cortona Player fully supports VRML97 and Java EAI. Figure 5 shows the structure of the application. Since the VRML client is only executable on a web browser, we used Microsoft Web Browser ActiveX Control to embed it into the application. Java Script [9, 10] served as the interface for transferring data between the VRML client and VC++. The structure of the application is shown in Fig. 5.

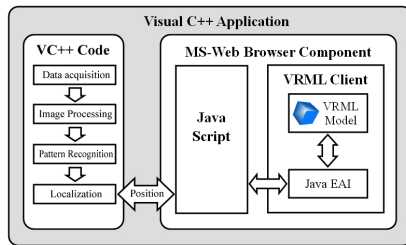


Fig. 5. The structure of the application

## 3 Path Finding

The search area is divided into a grid for simplicity and thus it becomes a simple 2-D array problem. Each item in the array represents one square of the grid, and its status was recorded as ‘walkable’ or ‘unwalkable’. The path was defined by determining which squares should be taken to proceed from A to B. Once the path was found, the

robot moves from the center of one square to the center of the next square until it reaches the target. These center points are called ‘nodes’ [17]. Selection of square involving the path is determined based on the following equation:

$$F = G + H \tag{5}$$

where G is the cost of moving from starting point A to a present position of robot, and H is the estimated cost moving from that the present position to the final destination. The variable H is often referred to as the heuristic since the values are estimated by designer’s judge. The actual cost is unknown until the path is decided, because there could be multiple obstacles in the way (e.g., walls, water, etc.). H can be estimated using a various methods. Each movement to a horizontal or vertical square was assigned a cost of 10, and a diagonal move cost 14. In Figure 6, the scores of F (top), G (bottom left), and H (bottom right) are shown in each square.

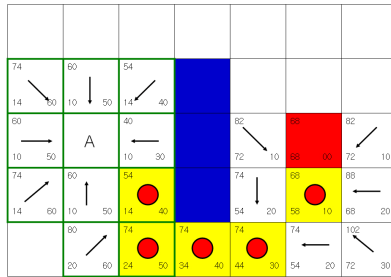


Fig. 6. The Robot’s path search

The path of robot was generated repeatedly by reviewing the grid and adding squares to the open list. The next square of the path was selected with the one has a lowest score in F and with a lowest scores in G also. For example, the square located on the right of A that has F score 40 in Fig. 6. And the score of G would be 20 if it assumed to be reached the next square marked 54 in F, in which the score of G is 14. Therefore, the square that has F score 40 can not be selected as the next square of the path even it has the lowest values in F. Selection of the next square should be one of two squares that have F = 54. We may choose the lower one in the grid (i.e., the yellow background, red dot) as a next square. After checking the adjacent squares, we can decide the square that has 54 as the right one for the path. The robot must travel directly downward, and then to the right to move around the corner. The squares below and to the left and directly below the current square are added to the open list, and the current square becomes their “parent.” This process is repeated until the target square is added to the closed list. In this way, a path of robot can be determined. In Figure 6, it can be traced by starting from the red target square, and working backwards moving from one square to its parent, following the arrows. This will eventually lead to the starting square, defining the path.

### 4 Simulation

For simulations, we constructed a miniature grid to substitute a real working environment and modeled the VRML scene accordingly. Landmarks were arranged at designated points, and the VRML scene was displayed in a wire-frame mode. The application took an image of the visual scene every 100 ms and processed all routines at the same time.

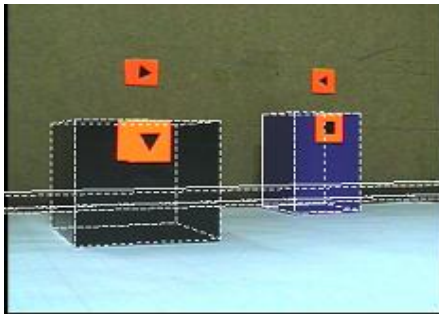


Fig. 7. VRML and vision overlay

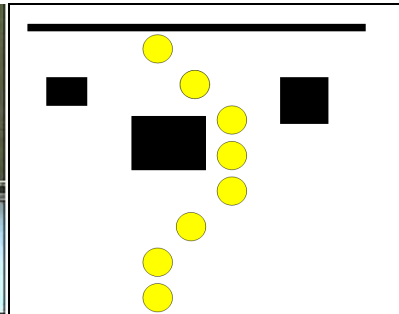


Fig. 8. Path of Robot in 2D

Landmarks were recognized using a neural network pattern recognition technique, and localization was performed at the same time. The overlay of the image obtained using the estimated position and angles and the real images are shown in Fig. 7. Exact localization will produce the exact images in the overlay. If the position and angle are not estimated properly, there exist mismatches between the real image and the VRML image. Fig. 8 is a path that will be used for tracking simulation. In the real implementation, the obstacles are modeled as rounded corners considering the size of the robot, thus the path tracking may not be the optimal one.

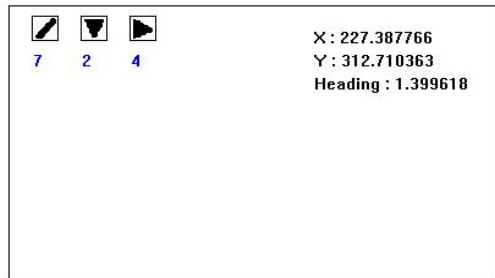
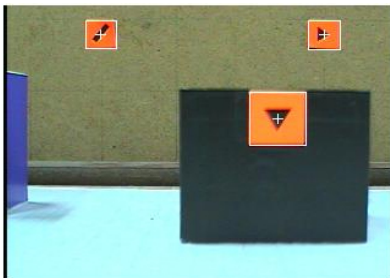


Fig. 9. x,y coordinates of Robot in simulation

Fig. 9 shows the images that the robot recognized the landmarks and performing the overlay of the VRML image and the real images. Fig. 10 shows the prototype robot performed path tracking using this result of Fig. 9.

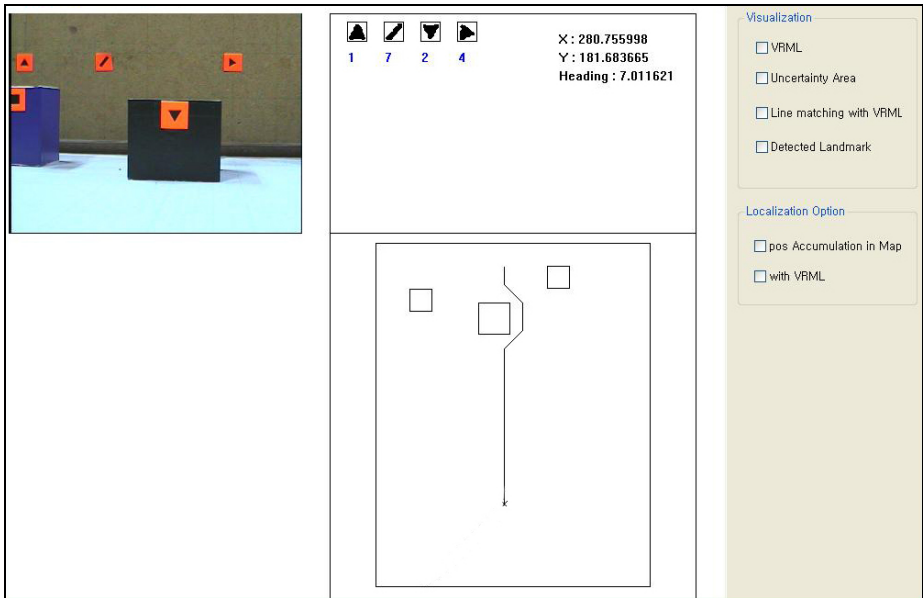


Fig. 10. Implementation of the system using a robot

## 5 Conclusion

In this paper, a method for self-localization and a path tracking have been proposed. The localization has been performed using a vision system and a Virtual Reality Modeling Language. A neural network was used to identifying the landmarks in the environment. The position and the angle of the robot are estimated using a linear method which will be realizable in real time operation. The error correction of the variables was done using the overlay of the VRML and the vision data. Finally, an experiment of self-positioning algorithm has been implemented to a prototype robot and also it performed autonomous path tracking successfully.

**Acknowledgements.** The authors are grateful for the support provided by the second stage of Brain Korea 21 project, Regional Research Center (R12-1998-026-02003-0) and Korean Electric Power Company (R-2004-B-12).

## References

1. Claus B. Madsen, Claus S. Andersen, "Optimal landmark selection for triangulation of robot position", *Robotics and Autonomous Systems*, vol. 23, Issue 4, pp. 277-292, July 1998.
2. K. Briechle and U. D. Hanebeck, Member, "Localization of a Mobile Robot Using Relative Bearing Measurements", *IEEE Transaction on Robotics and Automation*, vol. 20, no. 1, pp. 36-44 Feb 2004.

3. Eric Krotkov, "Mobile Robot Localization Using A Single Image", IEEE International Conference on Robotics and Automation, vol. 2, pp. 978-983, May 1989.
4. Margrit Betke and Leonid Gurvits, "Mobile Robot Localization Using Landmarks", IEEE Transaction on Robotics and Automation, vol. 13, no. 2, pp. 251-263 Apr 1997.
5. Esteves, J.S., Carvalho, A., Couto, C. "Generalized geometric triangulation algorithm for mobile robot absolute self-localization" Industrial Electronics, 2003. ISIE '03. 2003 IEEE International Symposium on, vol 1, pp. :346 – 351, 9-11 June 2003.
6. Cohen, Charles and Koss, Frank V., "A Comprehensive Study of Three Object Triangulation", Mobile Robots VII, SPIE vol. 1831, 1992.
7. D. J. Kang, J. E. Ha , "Digital Image Processing using Visual C++ ", SciTech, Korean, 2003.
8. <http://www.parallelgraphics.com/>
9. Alex, J., Vikramaditya, B., Nelson, B.J., "Teleoperated micromanipulation within a VRML environment using Java", Intelligent Robots and Systems, 1998. Proceedings, 1998 IEEE/RSJ International Conference on, vol. 3, pp. 1747-1752, 13-17 Oct. 1998.
10. Jiung-Yao Huang, "Increasing the visualization realism by frame synchronization between the VRML browser and the panoramic image viewer", International Journal of Human-Computer Studies, vol 55, Issue 3, pp. 311-336, Sep 2001.
11. Guilherme N. DeSouza and Avinash C. Kak, "Vision for Mobile Robot Navigation: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pp. 237-267, Feb 2002.
12. Wang, D, "Pattern Recognition: Neural Networks in Perspective", Expert, IEEE, vol 8, Issue 3, pp. 52-60, Aug 1993.
13. Antonios Gasteratos, Calos Beltran, Giorgio Metta, Giulio Sandini, "PRONTO:a system for mobile robot navigation via CAD-model guidance", Microprocessors and Microsystems, vol. 26, Issue 1, pp. 17-26, Feb 2002.
14. <http://www.policyalmanac.org> A star path finding for Beginners
15. Roland SIEGWART, Illah R. NOURBAKHS, "Introduce Autonomous mobile Robots"
16. Adam A. Razavian, Junping Sun "Cognitive Based Adaptive Path Planning Algorithm for Autonomous Robotic Vehicles", IEEE 2005.
17. <http://www.policyalmanac.org/>"A\* path finding for Beginners"

# Neural Network Control for Visual Guidance System of Mobile Robot

Young-Jae Ryoo

Department of Control System Engineering, Mokpo National University,  
61 Dorim-ri, Muan-goon, Jeonnam 534-729, Korea  
yjryoo@mokpo.ac.kr

**Abstract.** This paper describes a neural network control for a visual guidance system of a mobile robot to follow a guideline. Without complicated geometric reasoning from the image of a guideline to the robot-centered representation of a bird's eye view in conventional studies, the proposed system transfers the input of image information into the output of a steering angle directly. The neural network controller replaces the nonlinear relation of image information to a steering angle of robot on the real ground. For image information, the feature points of guideline are extracted from a camera image. In a straight and curved guideline, the driving performances by the proposed technology are measured in simulation and experimental test.

## 1 Introduction

Several visual guidance systems of mobile robot have been developed[1-8]. In the general mobile robot, the vision system acquires an image from a camera and uses a typical image processing algorithm to extract road line segments from the image. These line segments are transformed from the image coordinate system to the local vehicle coordinate system, and sent to the geometric reasoning module. Geometric reasoning uses local geometric supports and temporal continuity constraints to assign a consistent road interpretation to these line segments. The resulting road description is sent to lateral control system to generate a steering value and execute the mobile robot navigation. This system has difficulties of heavy calculation in given time; the geometric reasoning requires calculation of camera parameters and the lateral control depends on the parameters of road and robot. Given enough time, a sophisticated processing system might be able to overcome these difficulties. However the third challenging aspect of autonomous driving is that there is a limited amount of time available for processing sensor information. The system suffers from heavy computation needed to be complete within the given time. For the reasoning module, the computational complexity is governed by camera parameters while the lateral control being dependent on the parameters of the guideline and robot. Provided the processing time available is enough, this difficulty might be circumvented with a high speed computing system. Unfortunately, in practice, the processing time is limited, which demands more efficient control methodology from a computational standpoint.

Thus, in this paper, a fast control for a mobile robot with image sensors is proposed using neural network. In the proposed system, the position and orientation of a robot on a guideline are assumed to be unknown, but some points on the guideline are given in an image. The proposed system controls the robot so that it can follow the guidelines of guideline. The proposed intelligent system transfers the input of image information into the output of steering angle directly, without a complex geometric reasoning from an image to a robot-centered representation in conventional studies. The neural network system replaces the human driving skill of nonlinear relation between some points of guideline boundary on the camera image and the steering angle of robot on the real ground.

## 2 Neural Network Control for Visual Guidance

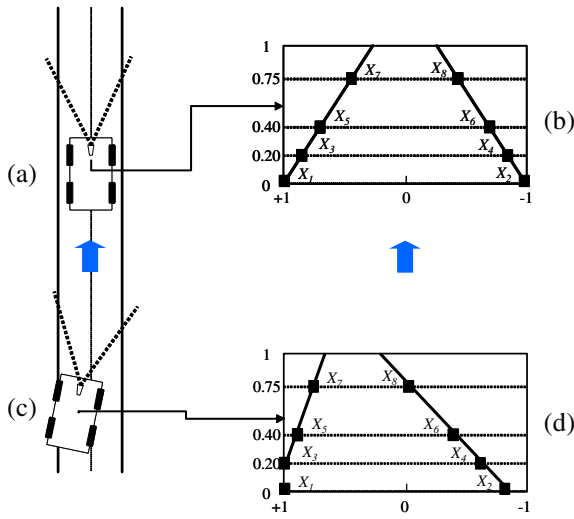
### 2.1 Feature Points on Image

The information of guideline boundary on image is obtained from the image processing of a guideline scene. As shown in Fig. 1, the geometrical relationship between the robot and the guideline can be described in image.

As depicted in Fig. 1, first, a camera captures an image from a road lane, from which some feature points can be extracted with image processing method. In Fig. 1, eight points are utilized as the representative of the road. Each image is scaled from 0 to 1 in the vertical axis and from  $-1$  to  $1$  in the horizontal axis, where 0 denotes the center. Positions on the vertical axis (0.0, 0.20, 0.40, and 0.75) make the horizontal line. Each horizontal line crosses with the road line and those crossed are indicated by  $(X_1, X_2, \dots, X_8)$ . In particular cases, any of the cross points may disappear on the image frame. For instance, the  $X_1$  in Fig. 1 does not cross the left lane. In this case, since it is hard to determine the missed crossover, the lateral position of the  $X_1$  is let be  $+1$  for convenience, thus the position of  $X_1$  becomes  $(+1, 0)$ .

The camera image contains the feature points on the guideline. With these points in the camera picture, the current position and orientation of the robot on the guideline can be uniquely determined by geometric calculations. On the basis of the above method, the following visual control method is introduced: Fig. 1 (d) is the current camera image, and Fig. 1 (b) is the desired camera image, obtained when the robot reaches the desired relative position and orientation on the guideline. The vision-based control system computes the error signals in terms of the lateral position and slope derived from the feature points in the visual image.

A steering angle generated by the proposed system makes that the feature points on the current image coincide with the desired image. The x-axis position of the feature point represents the relative position and orientation of the robot on the guideline. Then the robot is required to move its center to the lateral center of the guideline and to parallel the guideline by controlling its steering angle. It is significant that the feature points to the desired point in accordance with human's skill of driving.



**Fig. 1.** Relation of camera image of guideline view with the orientation and deviation of robot on the guideline, and feature points on camera image

**2.2 Neural Network Control**

The relation between the steering angle and the feature points on the camera image is a highly nonlinear function. Neural network is used for the nonlinear relation because it has the learning capability to map a set of input patterns to a set of output patterns. The inputs of the neural network ( $x_1, x_2, \dots, x_8$ ) are the x-axis position of the feature points ( $X_1, X_2, \dots, X_8$ ). The output of the neural network controller ( $y_0$ ) is the steering angle value for the robot ( $\delta$ ).

$$\begin{aligned}
 x_1 &= X_1 \\
 x_2 &= X_2 \\
 &\vdots \\
 x_8 &= X_8 \\
 y_0 &= \delta
 \end{aligned}
 \tag{1}$$

The learning algorithm of the neural network controller is back propagation. Learning data could be obtained from human’s driving. After the neural network controller learns the relation between input patterns and output patterns sufficiently, it makes a model of the relation between the position and orientation of the robot, and that of the guideline. Thus, a good model of the control task is obtained by learning, without inputting any knowledge about the specific robot and guideline.

**3 Computer Simulations**

To simulate in computer as shown in Fig. 2, robot model, transformation of coordinate system, and control algorithms should be determined. The used robot model is a



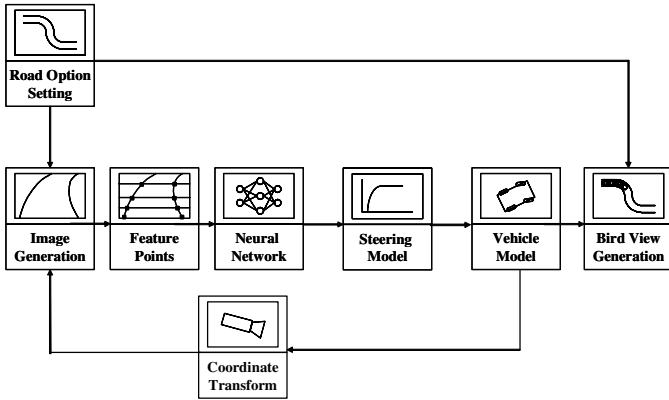


Fig. 2. Structure of simulation

general model, and has specific parameters. Through transformation of coordinate system, the guideline could be displayed on the camera image plane visually.

### 3.1 Mobile Robot Model

The general model of a robot with 4 wheels in the world coordinate system is shown in Fig. 3. The reference point  $(x_c^W, y_c^W)$  is located at the center point between the rear wheels. The heading angle  $\theta$  for  $X^W$ -axis of the world coordinate system and the steering angle  $\delta$  are defined in the robot coordinate system, and the model equation is expressed as follows;

$$\begin{aligned}
 \dot{\theta} &= \frac{l}{L_v} v \sin \delta \\
 \dot{x}_c^W &= v \cos \theta \cos \delta \\
 \dot{y}_c^W &= v \sin \theta \cos \delta.
 \end{aligned}
 \tag{2}$$

Since the robot coordinate system is used in control of mobile robots, the current position  $(x_c^W, y_c^W)$  of the robot in the world coordinate system is redefined as the point of origin for the robot coordinate system. When the robot which has the distance  $L_v$  between front wheel and rear wheel runs with velocity  $v$ , the new position of the robot is nonlinearly relative to steering angle  $\delta$  of front wheel and heading angle  $\theta$  determined by the robot direction and guideline direction.

### 3.2 Transformation from Ground to Image

In order to simulate in computer visually, the guideline has to be displayed on the camera image plane. Thus the coordinate transformations along the following steps are needed to determine the guideline of visual data from the guideline on the world coordinate system:

1) Transformation from the world coordinate system to the robot coordinate system. The position  $(x_c^w, y_c^w)$  on the world coordinates is redefined as the origin for the robot coordinates.

$$\begin{bmatrix} X^V \\ Y^V \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X^W - x_c^w \\ Y^W - y_c^w \end{bmatrix} \tag{3}$$

where  $(X^V, Y^V)$  is a point in the robot coordinate system,  $(X^W, Y^W)$  is a point in the world coordinate system,  $(x_c^w, y_c^w)$  is the reference point located at the center point between the rear wheels, and  $\theta$  is the heading angle of the robot.

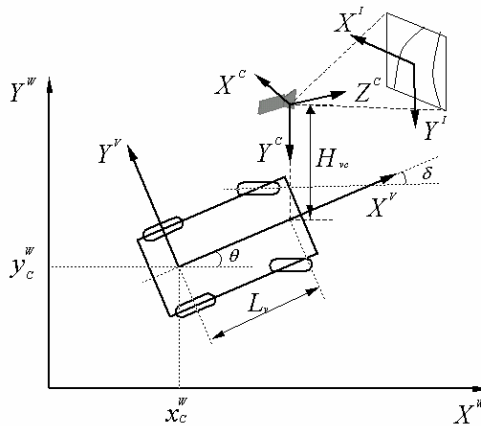


Fig. 3. Robot model and camera coordinate transformation

2) Transformation from the robot coordinate system to the camera coordinate system.

$$\begin{bmatrix} X^C \\ Y^C \\ Z^C \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} X^V \\ Y^V \end{bmatrix} + \begin{bmatrix} 0 \\ H_{vc} \\ L_{vc} \end{bmatrix} \tag{4}$$

where  $(X^C, Y^C, Z^C)$  is a point in the camera coordinate system,  $(X^V, Y^V)$  is a point on the robot coordinate system,  $H_{vc}$  is a height of the mounted camera from ground, and  $L_{vc}$  is the distance of the mounted camera from the center of the robot.

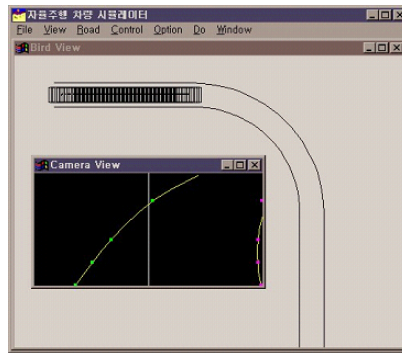
3) Transformation from the camera coordinate system to the image coordinate system.  $(\Theta, \Phi, \Psi)$  is the orientation of the camera with respect to the camera coordinates. The image plane  $(X^I, Y^I)$  corresponds to the  $X^C, Y^C$  plane at distance  $f$  (the focal length) from the position of the camera along the  $Z^C$  axis. The coordinate transformation is expressed by the rotation matrix  $R$  and the focal matrix  $F$ .

$$\begin{bmatrix} X^I \\ Y^I \end{bmatrix} = F \left\{ R \begin{bmatrix} X^C \\ Y^C \\ Z^C \end{bmatrix} \right\} \tag{5}$$

where  $(X_I, Y_I)$  is a point in the image coordinate system, and  $(X_C, Y_C, Z_C)$  is a point in the camera coordinate system.

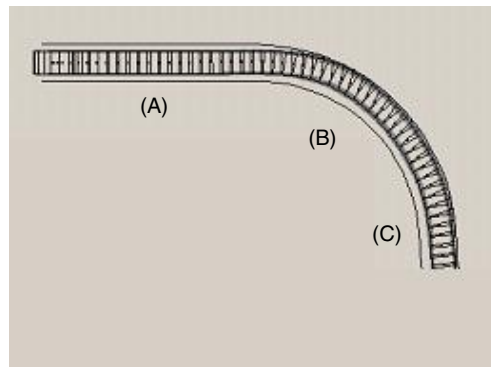
### 3.3 Simulation Results

Simulation results for the mobile robot are shown in this section. The simulation program is developed from programming of robot model, transformation of coordinate system, and control algorithms by C++ in IBM-PC.



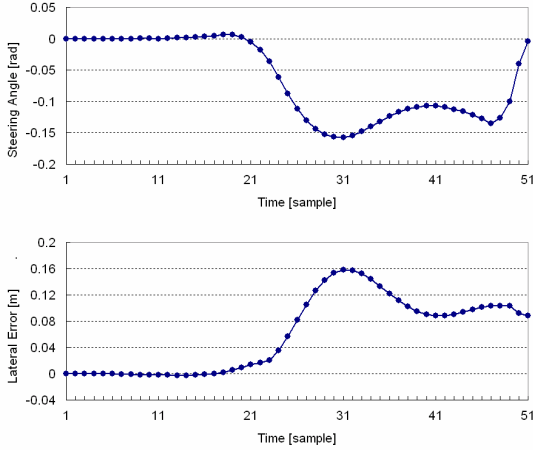
**Fig. 4.** The simulation program shows the camera image view and the bird's eye view

The performance of the proposed visual control system by neural network was evaluated using a robot driving on the track with a straight and curved guideline as shown in Fig. 5. Fig. 5 shows the bird's eye view of guideline map and the trajectories of robot's travel.



**Fig. 5.** Trajectories of robot driving on straight and curved guideline (straight line = 7meters, and curvature radius = 6 meters)

Fig. 6 shows the robot's steering angle during automatic guided driving in the guideline of Fig. 5. In Fig. 6, the steering angle is determined from neural network controller, and steering actuator model, and robot model. The steering angle is almost zero at straight guideline (section A), negative at right-turn curved guideline (section B), and returns to zero over curved guideline (section C). The controller steers the robot to left (about  $-0.16[\text{rad}]$ ) at right-turn curve (section B).



**Fig. 6.** Lateral error and steering angle

A lateral error is defined as distance between the center of guideline and the center of the robot. As shown in Fig. 6, automatic guided driving is completed in lateral error less than  $0.16[\text{m}]$ .

## 4 Experiments

### 4.1 Experimental Set-Up

The designed robot has 4 wheels, and its size is 1/3 of small passenger car. Driving torque comes from 2 induction motors set up at each rear wheel, and each motor has 200 watts. Front wheel is steered by worm gear with DC motor. Energy source is 4 batteries connected directly, and each battery has 6 volts. And CCD camera is used as a image sensor to get the guideline information. The control computer of robot has function to manage all systems, recognize the guideline direction from input camera image by guideline recognition algorithm, and make control signal of steering angle by neural network control. And the control computer manages and controls input information from various signal, also it inspects or watches the system state.

The computer is chosen personal computer for hardware extensibility and software flexibility. Electric system to control composes of vision system, steering control system, and speed control system. Vision system has a camera to acquire guideline image and image processing board IVP-150 to detect guideline boundary. Steering

control system has D/A converter to convert from control value to analog voltage, potentiometer to read the current steering angle, and analog P-controller. Speed control system has 16-bit counter to estimate current speed calculated from encoder pulses, D/A converter to convert command speed from controller to analog voltage, and inverter to drive 3-phase induction motor. Fig. 7 shows the designed robot.



Fig. 7. Designed robot

#### 4.2 Visual Guidance Test

The trajectory with a straight and curved guideline has the configuration as Fig. 5 evaluated in the simulation study. The guideline width is 1.2[m], the thickness of the guideline boundary is 0.05[m], and the total length of the guideline is set by about 10[m] merged the curved guideline. The curvature radius of the curved guideline is 6[m] as same as that of the computer simulation. Mobile robot is confirmed the useful driving performance on a straight guideline and curved guideline as shown Fig. 8.



Fig. 8. Visual guidance test on curved guideline. (a) at the starting (b) at the midway (c) at the finishing.

## 5 Conclusions

In this paper, a neural network control of a mobile robot with visual guidance system was described. The nonlinear relation between the image data and the control signals

for the steering angle can be learned by neural network. The validity of this neural network control was confirmed by computer simulations. This approach is effective because it essentially replaces human's skill of complex geometric calculations and image algorithm with a simple mapping of neural network system.

## References

1. Young-Jae Ryoo, Image Technology for Camera-Based Guided Vehicle, Lecture Notes in Computer Science 4319, Springer (2006) 1225-1233.
2. Young-Jae Ryoo, Vision-based Neuro-fuzzy Control of Autonomous Lane Following Vehicle, Neuro-fuzzy Pattern Recognition, World Scientific (2000) 249-264.
3. L. Zhai and C. E. Thorpe, Stereo and Neural Network-based Pedestrian Detection, IEEE Trans. on Intelligent Transportation System, Vol. 1, No. 3 (2000) 148-154
4. Dean A. Pomerleau, Neural Network Perception for Mobile Robot Guidance, Kluwer Academic Publishers (1997)
5. K. M. Passino, Intelligent Control for Autonomous Systems, IEEE spectrum, (1995) 55-62.
6. Sadayuki Tsugawa, Vision-based Vehicles in Japan: Machine Vision Systems and Driving Control systems", IEEE Transactions on Industrial Electronics, Vol. 41, No. 4 (1994) 398-405
7. J. Manigel and W. Leonhard, Vehicle Control by Computer Vision, IEEE Transactions on Industrial Electronics, Vol. 39, No. 3 (1992) 181-188.
8. Charles E. Thorpe, Vision and Navigation, the Carnegie Mellon Nablal, Kluwer Academic Publishers (1990)

# Cone-Realizations of Discrete-Time Systems with Delays

Tadeusz Kaczorek

Białystok Technical University,  
Faculty of Electrical Engineering,  
Wiejska 45D, 15-351 Białystok, Poland  
kaczorek@isep.pw.edu.pl

**Abstract.** A new notion of cone-realization for discrete-time linear systems with delays is proposed. Necessary and sufficient conditions for the existence of cone-realizations of discrete-time linear systems with delays are established. A procedure is proposed for computation of a cone-realization for a given proper rational matrix  $\mathbf{T}(z)$ . It is shown that there exists a  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization of  $\mathbf{T}(z)$  if and only if there exists a positive realization of  $\bar{\mathbf{T}}(z) = \mathbf{V}\mathbf{T}(z)\mathbf{Q}^{-1}$  where  $\mathbf{V}$ ,  $\mathbf{Q}$  and  $\mathbf{P}$  are non-singular matrices generating the cones  $\mathcal{V}$ ,  $\mathcal{Q}$  and  $\mathcal{P}$  respectively.

## 1 Introduction

In positive systems inputs, state variables and outputs take only non-negative values. Examples of positive systems are industrial processes involving chemical reactors, heat exchangers and distillation columns, storage systems, compartmental systems, water and atmospheric pollution models. A variety of models having positive linear systems behaviour can be found in engineering, management science, economics, social sciences, biology and medicine, etc.

Positive linear systems are defined on cones and not on linear spaces. Therefore, the theory of positive systems is more complicated and less advanced. An overview of state of the art in positive systems theory is given in the monographs [4, 5]. Recent developments in positive systems theory and some new results are given in [6]. Realizations problem of positive linear systems without time-delays has been considered in many papers and books [1, 4, 5].

Recently, the reachability, controllability and minimum energy control of positive linear discrete-time systems with time-delays have been considered in [3, 12]. The realization problem for positive multivariable discrete-time systems and continuous-time systems with delays was formulated and solved in [7, 8, 10, 11]. The notion of cone-realization of discrete-time systems without delays has been introduced in [9].

The main purpose of this paper is to present a method for computation of cone-realizations for a given proper rational transfer matrix of discrete-time systems with delays. Necessary and sufficient conditions will be established for the existence of cone realizations. A procedure will be proposed for computation of a cone-realization for a given proper rational transfer matrix.

To the best knowledge of the author the cone-positive realization problem for linear system has not been considered yet.

## 2 Preliminaries and Basic Definitions

Let  $\mathbb{R}^{m \times n}$  be the set of  $m \times n$  real matrices and  $\mathbb{R}^n = \mathbb{R}^{n \times 1}$ . The set of nonnegative integers will be denoted by  $\mathbb{Z}_+$ .

Consider the discrete-time linear system with delays

$$x_{i+1} = \mathbf{A}_0 x_i + \mathbf{A}_1 x_{i-1} + \mathbf{B}_0 u_i + \mathbf{B}_1 u_{i-1} \tag{1a}$$

$$y_i = \mathbf{C} x_i + \mathbf{D} u_i, \quad i \in \mathbb{Z}_+ = \{0, 1, \dots\} \tag{1b}$$

where  $x_i \in \mathbb{R}^n$ ,  $u_i \in \mathbb{R}^m$ ,  $y_i \in \mathbb{R}^p$  are the state, input and output vectors and  $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B}_0, \mathbf{B}_1 \in \mathbb{R}^{n \times m}$ ,  $\mathbf{C} \in \mathbb{R}^{p \times n}$ ,  $\mathbf{D} \in \mathbb{R}^{p \times m}$ .

**Definition 1.** Let  $\mathbf{P} = [p_1^T, \dots, p_n^T]^T \in \mathbb{R}^{n \times n}$  ( $T$  denotes the transpose) be nonsingular and  $p_k$  be the  $k$ th ( $k = 1, \dots, n$ ) its row. The set

$$\mathcal{P} := \left\{ x_i \in \mathbb{R}^n : \bigcap_{k=1}^n p_k x_i \geq 0 \right\} \tag{2}$$

is called a linear cone generated by the matrix  $\mathbf{P}$ .

In a similar way we may define for inputs  $u_i$  the linear cone

$$\mathcal{Q} := \left\{ u_i \in \mathbb{R}^m : \bigcap_{k=1}^m q_k u_i \geq 0 \right\} \tag{3}$$

generated by the nonsingular matrix  $\mathbf{Q} = [q_1^T, \dots, q_m^T]^T \in \mathbb{R}^{m \times m}$  and for outputs  $y_i$  the linear cone

$$\mathcal{V} := \left\{ y_i \in \mathbb{R}^p : \bigcap_{k=1}^p v_k y_i \geq 0 \right\} \tag{4}$$

generated by the nonsingular matrix  $\mathbf{V} = [v_1^T, \dots, v_p^T]^T \in \mathbb{R}^{p \times p}$

**Definition 2.** The linear system (1) is called  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -system if  $x_i \in \mathcal{P}$  and  $y_i \in \mathcal{V}$ ,  $i \in \mathbb{Z}_+$  for every  $x_0, x_{-1} \in \mathcal{P}$  and all  $u_{-1}, u_i \in \mathcal{Q}$ ,  $i \in \mathbb{Z}_+$ .

Let  $\mathbb{R}_+^{m \times n}$  be the set of  $m \times n$  real matrices with nonnegative entries. Note that for  $\mathcal{P} = \mathbb{R}_+^n$ ,  $\mathcal{Q} = \mathbb{R}_+^m$ ,  $\mathcal{V} = \mathbb{R}_+^p$  we obtain  $(\mathbb{R}_+^n, \mathbb{R}_+^m, \mathbb{R}_+^p)$  - positive linear system (shortly positive system) which is equivalent to the classical positive system [7, 8, 10, 11].

**Theorem 1.** The linear system (1) is  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -system if and only if

$$\begin{aligned} \bar{\mathbf{A}}_k &= \mathbf{P} \mathbf{A}_k \mathbf{P}^{-1} \in \mathbb{R}_+^{n \times n}, \quad \bar{\mathbf{B}}_k = \mathbf{P} \mathbf{B}_k \mathbf{Q}^{-1} \in \mathbb{R}_+^{n \times m}, \quad k = 0, 1, \\ \bar{\mathbf{C}} &= \mathbf{V} \mathbf{C} \mathbf{P}^{-1} \in \mathbb{R}_+^{p \times n}, \quad \bar{\mathbf{D}} = \mathbf{V} \mathbf{D} \mathbf{Q}^{-1} \in \mathbb{R}_+^{p \times m} \end{aligned} \tag{5}$$

**Proof.** Let

$$\bar{x}_i = \mathbf{P} x_i, \quad \bar{u}_i = \mathbf{Q} u_i, \quad \text{and} \quad \bar{y}_i = \mathbf{V} y_i \tag{6}$$



From Definition 1 it follows that if  $x_i \in \mathcal{P}$  then  $\bar{x}_i \in \mathbb{R}_+^n$  if  $u_i \in \mathcal{Q}$  then  $\bar{u}_i \in \mathbb{R}_+^m$  and if  $y_i \in \mathcal{V}$  then  $\bar{y}_i \in \mathbb{R}_+^p$ .

From (1) and (6) we have

$$\begin{aligned} \bar{x}_{i+1} &= \mathbf{P}x_{i+1} = \mathbf{P}\mathbf{A}_0x_i + \mathbf{P}\mathbf{A}_1x_{i-1} + \mathbf{P}\mathbf{B}_0u_i + \mathbf{P}\mathbf{B}_1u_{i-1} = \\ &= \mathbf{P}\mathbf{A}_0\mathbf{P}^{-1}\bar{x}_i + \mathbf{P}\mathbf{A}_1\mathbf{P}^{-1}\bar{x}_{i-1} + \mathbf{P}\mathbf{B}_0\mathbf{Q}^{-1}\bar{u}_i + \mathbf{P}\mathbf{B}_1\mathbf{Q}^{-1}\bar{u}_{i-1} = \\ &= \bar{\mathbf{A}}_0\bar{x}_i + \bar{\mathbf{A}}_1\bar{x}_{i-1} + \bar{\mathbf{B}}_0\bar{u}_i + \bar{\mathbf{B}}_1\bar{u}_{i-1} \end{aligned} \tag{7a}$$

and

$$\bar{y}_i = \mathbf{V}y_i = \mathbf{V}\mathbf{C}x_i + \mathbf{V}\mathbf{D}u_i = \mathbf{V}\mathbf{C}\mathbf{P}^{-1}\bar{x}_i + \mathbf{V}\mathbf{D}\mathbf{Q}^{-1}\bar{u}_i = \bar{\mathbf{C}}\bar{x}_i + \bar{\mathbf{D}}\bar{u}_i \tag{7b}$$

It is well-known [8] that the system (7) is the positive system if and only if the conditions (5) are satisfied. ■

**Lemma.** The transfer matrix

$$\mathbf{T}(z) = \mathbf{C} \left[ \mathbf{I}_n z^2 - \mathbf{A}_0 z - \mathbf{A}_1 \right]^{-1} (\mathbf{B}_0 z + \mathbf{B}_1) + \mathbf{D} \tag{8}$$

of the  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -system (1) and the transfer matrix

$$\bar{\mathbf{T}}(z) = \bar{\mathbf{C}} \left[ \mathbf{I}_n z^2 - \bar{\mathbf{A}}_0 z - \bar{\mathbf{A}}_1 \right]^{-1} (\bar{\mathbf{B}}_0 z + \bar{\mathbf{B}}_1) + \bar{\mathbf{D}} \tag{9}$$

of the positive system (1) are related by the equality

$$\bar{\mathbf{T}}(z) = \mathbf{V}\mathbf{T}(z)\mathbf{Q}^{-1} \tag{10}$$

**Proof.** Using (9), (5) and (8) we obtain

$$\begin{aligned} \bar{\mathbf{T}}(z) &= \bar{\mathbf{C}} \left[ \mathbf{I}_n z^2 - \bar{\mathbf{A}}_0 z - \bar{\mathbf{A}}_1 \right]^{-1} (\bar{\mathbf{B}}_0 z + \bar{\mathbf{B}}_1) + \bar{\mathbf{D}} = \\ &= \mathbf{V}\mathbf{C}\mathbf{P}^{-1} \left[ \mathbf{I}_n z^2 - \mathbf{P}(\mathbf{A}_0 z + \mathbf{A}_1)\mathbf{P}^{-1} \right]^{-1} \mathbf{P}(\mathbf{B}_0 z + \mathbf{B}_1)\mathbf{Q}^{-1} + \mathbf{V}\mathbf{D}\mathbf{Q}^{-1} = \\ &= \mathbf{V}\mathbf{C}\mathbf{P}^{-1} \left[ \mathbf{P}(\mathbf{I}_n z^2 - \mathbf{A}_0 z - \mathbf{A}_1)\mathbf{P}^{-1} \right]^{-1} \mathbf{P}(\mathbf{B}_0 z + \mathbf{B}_1)\mathbf{Q}^{-1} + \mathbf{V}\mathbf{D}\mathbf{Q}^{-1} = \\ &= \mathbf{V}\mathbf{C} \left[ \mathbf{I}_n z^2 - \mathbf{A}_0 z - \mathbf{A}_1 \right]^{-1} (\mathbf{B}_0 z + \mathbf{B}_1)\mathbf{Q}^{-1} + \mathbf{V}\mathbf{D}\mathbf{Q}^{-1} = \mathbf{V}\mathbf{T}(z)\mathbf{Q}^{-1}. \end{aligned} \quad \square$$

### 3 Problem Formulation

Consider the linear system (1) with its transfer matrix (8). Let  $\mathbb{R}^{p \times m}(z)$  be the set of  $p \times m$  rational proper matrices.

**Definition 3.** Matrices  $\mathbf{A}_k \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B}_k \in \mathbb{R}^{n \times m}$ ,  $k = 0, 1$ ,  $\mathbf{C} \in \mathbb{R}^{p \times n}$ ,  $\mathbf{D} \in \mathbb{R}^{p \times m}$  are called a  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization of a given proper  $\mathbf{T}(z) \in \mathbb{R}^{p \times m}(z)$  if they satisfy the equality (8) and the conditions

$$\begin{aligned} \mathbf{P}\mathbf{A}_k\mathbf{P}^{-1} &\in \mathbb{R}_+^{n \times n}, \quad \mathbf{P}\mathbf{B}_k\mathbf{Q}^{-1} \in \mathbb{R}_+^{n \times m}, \quad k = 0, 1, \\ \mathbf{V}\mathbf{C}\mathbf{P}^{-1} &\in \mathbb{R}_+^{p \times n}, \quad \mathbf{V}\mathbf{D}\mathbf{Q}^{-1} \in \mathbb{R}_+^{p \times m} \end{aligned} \tag{11}$$

where  $\mathbf{P}$ ,  $\mathbf{Q}$  and  $\mathbf{V}$  are nonsingular matrices generating the cones  $\mathcal{P}$ ,  $\mathcal{Q}$  and  $\mathcal{V}$ , respectively.

The  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization problem can be stated as follows: Given a proper rational matrix  $\mathbf{T}(z) \in \mathbb{R}^{p \times m}(z)$  and non-singular matrices  $\mathbf{P}$ ,  $\mathbf{Q}$ ,  $\mathbf{V}$  generating the cones  $\mathcal{P}$ ,  $\mathcal{Q}$  and  $\mathcal{V}$  find a  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization of  $\mathbf{T}(z)$

A procedure for computation of a  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization of  $\mathbf{T}(z)$  will be proposed and solvability conditions of the problem will be established.

## 4 Problem Solution

### 4.1 Computation of Positive Realizations

In this section the sufficient conditions for the existence and the procedure for computation of the positive realization of  $\bar{\mathbf{T}}(z)$  will be presented.

From (9) we have

$$\bar{\mathbf{D}} = \lim_{z \rightarrow \infty} \bar{\mathbf{T}}(z) \tag{12}$$

since  $\lim_{z \rightarrow \infty} [z^{-1}(\mathbf{I}_n z^2 - \bar{\mathbf{A}}_0 z - \bar{\mathbf{A}}_1)]^{-1} = 0$ .

The strictly proper part of  $\mathbf{T}(z)$  is given by

$$\bar{\mathbf{T}}_{sp}(z) = \bar{\mathbf{T}}(z) - \bar{\mathbf{D}} \tag{13}$$

In [8] it was shown that if the matrices  $\mathbf{A}_0$  and  $\mathbf{A}_1$  have the following forms

$$\mathbf{A}_0 = \begin{bmatrix} 0 & \dots & 0 & a_1 \\ 0 & \dots & 0 & a_3 \\ 0 & \dots & 0 & a_5 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & a_{2n-1} \end{bmatrix} \in \mathbb{R}^{n \times n}, \mathbf{A}_1 = \begin{bmatrix} 0 & 0 & \dots & 0 & a_0 \\ 1 & 0 & \dots & 0 & a_2 \\ 0 & 1 & \dots & 0 & a_4 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & a_{2(n-1)} \end{bmatrix} \in \mathbb{R}^{n \times n} \tag{14}$$

then

$$d(z) = \det[\mathbf{I}_n z^2 - \mathbf{A}_0 z - \mathbf{A}_1] = z^{2n} - a_{2n-1} z^{2n-1} - \dots - a_1 z - a_0 \tag{15}$$

and the  $n$ th row of the adjoint matrix  $\text{Adj}[\mathbf{I}_n z^2 - \mathbf{A}_0 z - \mathbf{A}_1]$  has the form

$$\mathbf{R}_n(z) = [1 \quad z^2 \quad z^4 \quad \dots \quad z^{2(n-1)}] \tag{16}$$

The strictly proper  $\bar{\mathbf{T}}_{sp}(z)$  can be always written in the form

$$\bar{\mathbf{T}}_{sp} = \begin{bmatrix} \mathbf{N}_1^T(z) & & & \\ d_1(z) & \dots & & \mathbf{N}_p^T(z) \\ & & & d_p(z) \end{bmatrix} \tag{17}$$

where

$$d_i(z) = z^{2q_i} - a_{i2q_i-1}z^{2q_i-1} - \dots - a_{i1}z - a_{i0}, \quad i = 1, \dots, p \tag{18}$$

is the least common denominator of the  $i$ th row of  $\bar{\mathbf{T}}_{sp}(z)$  and

$$\begin{aligned} \mathbf{N}_i(z) &= [n_{i1}(z) \quad n_{i2}(z) \quad \dots \quad n_{im}(z)], \quad i = 1, \dots, p \\ n_{ij}(z) &= n_{ij}^{2q_i-1}z^{2q_i-1} + \dots + n_{ij}^1z + n_{ij}^0, \quad j = 1, \dots, m \end{aligned} \tag{19}$$

To the polynomial (18) we associate the pair of the matrices

$$\bar{\mathbf{A}}_{0i} = \begin{bmatrix} 0 & \dots & 0 & a_{i1} \\ 0 & \dots & 0 & a_{i3} \\ \dots & \ddots & \dots & \dots \\ 0 & \dots & 0 & a_{i2q_i-1} \end{bmatrix} \in \mathbb{R}^{q_i \times q_i}, \quad \bar{\mathbf{A}}_{1i} = \begin{bmatrix} 0 & 0 & \dots & 0 & a_{i0} \\ 1 & 0 & \dots & 0 & a_{i2} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & a_{i2(q_i-1)} \end{bmatrix} \in \mathbb{R}^{q_i \times q_i}, \tag{20}$$

$$i = 1, \dots, p$$

satisfying the condition

$$d_i(z) = \det [\mathbf{I}_{q_i} z^2 - \bar{\mathbf{A}}_{0i} z - \bar{\mathbf{A}}_{1i}], \quad i = 1, \dots, p \tag{21}$$

Let

$$\bar{\mathbf{C}} = \text{block diag} [\bar{\mathbf{C}}_1 \quad \bar{\mathbf{C}}_2 \quad \dots \quad \bar{\mathbf{C}}_p], \quad \bar{\mathbf{C}}_i = [0 \quad 0 \quad \dots \quad 1] \in \mathbb{R}^{1 \times q_i}, \quad i = 1, \dots, p \tag{22}$$

and

$$\begin{bmatrix} b_{1j}^k \\ \vdots \\ b_{pj}^k \end{bmatrix}, \quad b_{ij}^k = \begin{bmatrix} b_{ij}^{k1} \\ \vdots \\ b_{ij}^{kq_i} \end{bmatrix}, \quad k = 0, 1; i = 1, \dots, p; j = 1, \dots, m \tag{23}$$

be the  $j$ th column of the matrix  $\bar{\mathbf{B}}_k$  ( $k = 0, 1$ ).

In [8] it was shown that the entries of  $\mathbf{B}_k$ ,  $k = 0, 1$  are given by

$$\begin{aligned} b_{1j}^{0q_1} &= n_{1j}^{2q_1-1}, b_{1j}^{1q_1} = n_{1j}^{2(q_1-1)}, \dots, b_{1j}^{01} = n_{1j}^1, b_{1j}^{11} = n_{1j}^0, \quad j = 1, \dots, m \\ &\dots \dots \dots \tag{24} \\ b_{pj}^{0q_p} &= n_{pj}^{2q_p-1}, b_{pj}^{1q_p} = n_{pj}^{2(q_p-1)}, \dots, b_{pj}^{01} = n_{pj}^1, b_{pj}^{11} = n_{pj}^0 \end{aligned}$$

**Theorem 2.** [8] There exists a positive realization  $\bar{\mathbf{A}}_k \in \mathbb{R}_+^{n \times n}$ ,  $\bar{\mathbf{B}}_k \in \mathbb{R}_+^{n \times m}$ ,  $\bar{\mathbf{C}} \in \mathbb{R}_+^{p \times n}$ ,  $\bar{\mathbf{D}} \in \mathbb{R}_+^{p \times m}$  of  $\bar{\mathbf{T}}(z)$  if

(i)

$$\bar{\mathbf{T}}(\infty) = \lim_{z \rightarrow \infty} (\bar{\mathbf{T}}(z)) \in \mathbb{R}_+^{p \times m} \tag{25}$$

(ii) the coefficients of  $d_i(z)$ ,  $i = 1, \dots, p$  are nonnegative, i.e.

$$a_{ij} \geq 0 \text{ for } i = 1, \dots, p; j = 0, 1, \dots, 2q_i - 1 \tag{26a}$$

(iii)  $n_{ij}^k \geq 0$  for  $i = 0, 1, \dots, p; j = 1, \dots, m; k = 0, 1, \dots, 2q_i - 1$ .

If the conditions (25) and (26) are satisfied then a positive realization of  $\bar{\mathbf{T}}(z)$  can be computed by the use of the following procedure.

**Procedure 1**

**Step 1.** Using (12) and (13) find  $\bar{\mathbf{D}} \in \mathbb{R}_+^{p \times m}$  and the strictly proper matrix  $\bar{\mathbf{T}}_{sp}(z)$ .

**Step 2.** Knowing the coefficients  $a_{ij}$  ( $i = 0, 1, \dots, p; j = 0, 1, \dots, 2q_i - 1$ ) of  $d_i(z)$   $i = 1, \dots, p$  find the matrices (20) and

$$\begin{aligned} \bar{\mathbf{A}}_0 &= \text{block diag} [\bar{\mathbf{A}}_{01}, \dots, \bar{\mathbf{A}}_{0p}] \in \mathbb{R}_+^{n \times n}, \\ \bar{\mathbf{A}}_1 &= \text{block diag} [\bar{\mathbf{A}}_{11}, \dots, \bar{\mathbf{A}}_{1p}] \in \mathbb{R}_+^{n \times n} \end{aligned} \tag{27}$$

**Step 3.** Knowing the coefficients  $n_{ij}^k$ , ( $i = 0, 1, \dots, p; j = 1, \dots, m; k = 0, 1, \dots, 2q_i - 1$ ) of  $\mathbf{N}_i(z)$  ( $i = 1, \dots, p$ ) and using (24) find  $\bar{\mathbf{B}}_k$  for  $k = 0, 1$  and the matrix  $\bar{\mathbf{C}}$  of the form (22).

**4.2 Computation of  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -Cone Realizations**

A  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization for a given  $\mathbf{T}(z) \in \mathbb{R}^{p \times m}(z)$  and non-singular matrices  $\mathbf{P}$ ,  $\mathbf{Q}$ ,  $\mathbf{V}$  can be computed by the use of the following procedure.

**Procedure 2**

**Step 1.** Knowing  $\mathbf{T}(z)$  and the matrices  $\mathbf{V}$ ,  $\mathbf{Q}$  and using (10) compute the transfer matrix  $\bar{\mathbf{T}}(z)$ ,

**Step 2.** Using Procedure 1 compute a positive realization  $\bar{\mathbf{A}}_k$ ,  $\bar{\mathbf{B}}_k$ ,  $k = 0, 1$ ,  $\bar{\mathbf{C}}$ ,  $\bar{\mathbf{D}}$  of the transfer matrix  $\bar{\mathbf{T}}(z)$

**Step 3.** Using the relations

$$\mathbf{A}_k = \mathbf{P}\bar{\mathbf{A}}_k\mathbf{P}^{-1}, \mathbf{B}_k = \mathbf{P}^{-1}\bar{\mathbf{B}}_k\mathbf{Q}, k = 0, 1, \mathbf{C} = \mathbf{V}^{-1}\bar{\mathbf{C}}\mathbf{P}, \mathbf{D} = \mathbf{V}^{-1}\bar{\mathbf{D}}\mathbf{Q} \tag{28}$$

compute the desired realization.

The procedure follows from Lemma and the realizations (5).

**Theorem 3.** There exists a  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization of  $\mathbf{T}(z)$  if and only if there exists a positive realization of  $\bar{\mathbf{T}}(z)$ .

The proof follows immediately from Procedure 2 and Lemma.

From Theorem 2 for single input single-output system ( $m = p = 1$ ) we have the following important corollary.

**Corollary.** There exists a  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization  $\mathbf{A}_k$ ,  $\mathbf{B}_k$ ,  $k = 0, 1$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  of the transfer function  $\mathbf{T}(z)$  if and only if there exist a positive  $\mathbf{A}_k$ ,  $\mathbf{B}_k$ ,  $k = 0, 1$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  of  $\bar{\mathbf{T}}(z)$  and the realization are related by

$$\begin{aligned} \mathbf{A}_k &= \mathbf{P}\bar{\mathbf{A}}_k\mathbf{P}^{-1}, \quad \mathbf{B}_k = \mathbf{P}^{-1}\bar{\mathbf{B}}_k\mathbf{Q}, \quad k = 0, 1, \quad \mathbf{C} = \bar{\mathbf{C}}\mathbf{P} \\ &\text{(or } \mathbf{B}_k = \mathbf{P}^{-1}\bar{\mathbf{B}}_k \text{ and } \mathbf{C} = \mathbf{k}\bar{\mathbf{C}}\mathbf{P}), \quad \mathbf{D} = \mathbf{k}\bar{\mathbf{D}} \end{aligned} \tag{29}$$

where  $\mathbf{k} = \mathbf{Q}\mathbf{V}^{-1}$  is a scalar.

For  $m = p = 1$  the transfer functions  $\bar{\mathbf{T}}(z)$  and  $\mathbf{T}(z)$  related by  $\bar{\mathbf{T}}(z) = \mathbf{k}\mathbf{T}(z)$  ( $\mathbf{k} = \mathbf{Q}\mathbf{V}^{-1}$ ).

## 5 Examples

### 5.1 Example 1

Given

$$T(z) = \frac{3z^3 + z^2 + z + 2}{z^4 - 2z^3 - 3z^2 - z - 2} \tag{30}$$

and

$$\mathbf{P} = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{Q} = \mathbf{V} = 1 \tag{31}$$

find a  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization of (30).

The  $\mathcal{P}$ -cone generated by the matrix (31) is shown in Fig. 1.

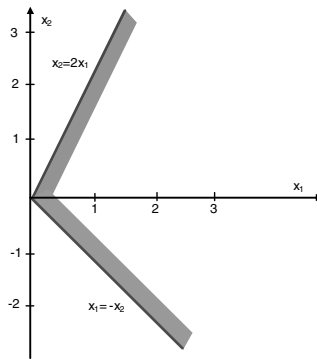


Fig. 1.

Using Procedure 2 we obtain

**Step 2.1.** In this case  $\bar{\mathbf{T}}(z) = \mathbf{T}(z)$  since  $\mathbf{Q} = \mathbf{V} = 1$

**Step 2.2.** Using Procedure 1 we obtain a positive realization of (30) of the form

$$\bar{\mathbf{A}}_0 = \begin{bmatrix} 0 & 1 \\ 0 & 2 \end{bmatrix}, \quad \bar{\mathbf{A}}_1 = \begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix}, \quad \bar{\mathbf{B}}_0 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \quad \bar{\mathbf{B}}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad \bar{\mathbf{C}} = [0 \quad 1], \quad \bar{\mathbf{D}} = 0 \tag{32}$$

**Step 2.3.** Using (28), (32) and (31) we obtain the desired realization in the form

$$\begin{aligned} \mathbf{A}_0 &= \mathbf{P}^{-1} \bar{\mathbf{A}}_0 \mathbf{P} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \mathbf{A}_1 = \mathbf{P}^{-1} \bar{\mathbf{A}}_1 \mathbf{P} = \frac{1}{3} \begin{bmatrix} 7 & 4 \\ 8 & 2 \end{bmatrix}, \mathbf{B}_0 = \mathbf{P}^{-1} \bar{\mathbf{B}}_0 \mathbf{Q} = \frac{1}{3} \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \\ \mathbf{B}_1 &= \mathbf{P}^{-1} \bar{\mathbf{B}}_1 \mathbf{Q} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{C} = \mathbf{V}^{-1} \bar{\mathbf{C}} \mathbf{P} = [4 \quad 1], \quad \mathbf{D} = \mathbf{V}^{-1} \bar{\mathbf{D}} \mathbf{Q} = 0 \end{aligned} \quad (33)$$

## 5.2 Example 2

Given the transfer matrix

$$\begin{aligned} \mathbf{T}(z) &= \frac{1}{2(z^2 - 2z - 1)(z^4 - 2z^3 - z^2 - z - 2)} \times \\ &\begin{bmatrix} -2z^6 + 7z^5 + z^4 - 10z^3 - 8z^2 - 10z - 5 & , 7z^6 - 23z^5 + 7z^4 + 11z^3 + z^2 + 23z + 10 \\ z^5 - 3z^4 + 4z^2 - 2z - 3 & , -3z^6 + 13z^5 - 9z^4 - 9z^3 - 5z^2 - 17z - 6 \end{bmatrix} \end{aligned} \quad (34)$$

and the non-singular matrices

$$\mathbf{P} = \begin{bmatrix} 1 & -2 & 0 \\ -1 & 1 & 2 \\ 1 & -1 & 0 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad (35)$$

find a  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization of (34).

In this case  $m = p = 2$ . Using Procedure 2 we obtain

**Step 2.1.** From (10) and (34) we have

$$\bar{\mathbf{T}}(z) = \mathbf{V} \mathbf{T}(z) \mathbf{Q}^{-1} = \begin{bmatrix} \frac{z^4 - 2z^3 + z^2 - 2}{z^4 - 2z^3 - z^2 - z - 2} & \frac{2z^4 - 3z^3 - 2z^2 - 2z - 3}{z^4 - 2z^3 - z^2 - z - 2} \\ \frac{z + 1}{z^2 - 2z - 1} & \frac{z^2 - z - 1}{z^2 - 2z - 1} \end{bmatrix} \quad (36)$$

**Step 2.2.** To find a positive realization of (36) we use Procedure 1.

It is easy to verify that the assumptions of Theorem 2 for the transfer matrix (36) are satisfied.

Using the Procedure 1 we obtain the following

**Step 1.1.** From (12) and (36) we have

$$\bar{\mathbf{D}} = \lim_{z \rightarrow \infty} \bar{\mathbf{T}}(z) = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \quad (37)$$

and

$$\begin{aligned} \bar{\mathbf{T}}_{sp}(z) &= \mathbf{T}(z) - \bar{\mathbf{D}} = \\ &= \begin{bmatrix} \frac{2z^2 + z}{z^4 - 2z^3 - z^2 - z - 2} & \frac{z^3 + 1}{z^4 - 2z^3 - z^2 - z - 2} \\ \frac{z + 1}{z^2 - 2z - 1} & \frac{z}{z^2 - 2z - 1} \end{bmatrix} = \begin{bmatrix} \frac{n_{11}(z)}{d_1(z)} & \frac{n_{12}(z)}{d_1(z)} \\ \frac{n_{21}(z)}{d_2(z)} & \frac{n_{22}(z)}{d_2(z)} \end{bmatrix} \end{aligned} \tag{38}$$

**Step 1.2.** Taking into account that in this case  $d_1(z) = z^4 - 2z^3 - z^2 - z - 2$ ,  $d_2(z) = z^2 - 2z - 1$  and using (20) and (27) we obtain

$$\bar{\mathbf{A}}_0 = \begin{bmatrix} \bar{\mathbf{A}}_{01} & 0 \\ 0 & \bar{\mathbf{A}}_{02} \end{bmatrix} = \begin{bmatrix} 0 & 1 & | & 0 \\ 0 & 2 & | & 0 \\ 0 & 0 & | & 2 \end{bmatrix}, \quad \bar{\mathbf{A}}_1 = \begin{bmatrix} \bar{\mathbf{A}}_{11} & 0 \\ 0 & \bar{\mathbf{A}}_{12} \end{bmatrix} = \begin{bmatrix} 0 & 2 & | & 0 \\ 1 & 1 & | & 0 \\ 0 & 0 & | & 1 \end{bmatrix} \tag{39}$$

**Step 1.3.** In this case  $n_{11} = 2z^2 + z$ ,  $n_{12} = z^3 + 1$ ,  $n_{21} = z + 1$ ,  $n_{22} = z$ . Using (24) we obtain

$$\begin{aligned} \bar{\mathbf{B}}_0 &= \begin{bmatrix} b_{11}^{01} & b_{12}^{01} \\ b_{11}^{02} & b_{12}^{02} \\ b_{21}^{01} & b_{22}^{01} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad \bar{\mathbf{B}}_1 = \begin{bmatrix} b_{11}^{11} & b_{12}^{11} \\ b_{11}^{12} & b_{12}^{12} \\ b_{21}^{11} & b_{22}^{11} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 2 & 0 \\ 1 & 0 \end{bmatrix}, \\ \text{and } \bar{\mathbf{C}} &= \begin{bmatrix} \bar{C}_1 & 0 \\ 0 & \bar{C}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{40}$$

The desired positive realization of (36) is given by (37), (39) and (40).

**Step 2.3.** Using (28) and (39) we obtain the desired realization in the form

$$\begin{aligned} \mathbf{A}_0 &= \mathbf{P}^{-1} \bar{\mathbf{A}}_0 \mathbf{P} = \begin{bmatrix} 5 & -5 & -2 \\ 3 & -3 & -2 \\ 0 & 0 & 2 \end{bmatrix}, \quad \mathbf{A}_1 = \mathbf{P}^{-1} \bar{\mathbf{A}}_1 \mathbf{P} = \begin{bmatrix} 4 & -4 & -4 \\ 3 & -3 & -4 \\ 0.5 & -1 & 1 \end{bmatrix} \\ \mathbf{B}_0 &= \mathbf{P}^{-1} \bar{\mathbf{B}}_0 \mathbf{Q} = \begin{bmatrix} -1 & 5 \\ -1 & 2 \\ -0.5 & 2.5 \end{bmatrix}, \quad \mathbf{B}_1 = \mathbf{P}^{-1} \bar{\mathbf{B}}_1 \mathbf{Q} = \begin{bmatrix} 3 & 0 \\ 2 & -1 \\ 1.5 & 1.5 \end{bmatrix}, \\ \mathbf{C} &= \mathbf{V}^{-1} \bar{\mathbf{C}} \mathbf{P} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix}, \quad \mathbf{D} = \mathbf{V}^{-1} \bar{\mathbf{D}} \mathbf{Q} = \frac{1}{2} \begin{bmatrix} -2 & 7 \\ 0 & -3 \end{bmatrix} \end{aligned}$$

## 6 Concluding Remarks

The notions of a  $\mathcal{P}$ -cone generated by a non-singular matrix  $\mathbf{P}$  has been introduced (Definition 1) and of a  $(\mathcal{P}, \mathbf{Q}, \mathcal{V})$ -system. Necessary and sufficient condition for a system to be a  $(\mathcal{P}, \mathbf{Q}, \mathcal{V})$ -system for discrete-time linear systems have been established

(Theorem 1). A procedure has been proposed for computation of a  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization for a given proper rational matrix  $\mathbf{T}(z)$ . It has been shown (Theorem 2) that there exist a  $(\mathcal{P}, \mathcal{Q}, \mathcal{V})$ -cone realization of  $\mathbf{T}(z)$  if and only if there exist a positive realization of  $\bar{\mathbf{T}}(z) = \mathbf{V}\mathbf{T}(z)\mathbf{Q}^{-1}$ , where  $\mathbf{V}$ ,  $\mathbf{Q}$  are non-singular matrices generating the cones  $\mathcal{V}$  and  $\mathcal{Q}$  respectively. The procedure has been illustrated by two numerical examples. The considerations can be extended for continuous-time linear systems and multivariable discrete-time systems with delays. Using the notion of  $\mathcal{P}$ -cone we may introduce the  $\mathcal{P}$ -cone reachability,  $\mathcal{P}$ -cone controllability and  $\mathcal{P}$ -cone observability of discrete-time and continuous-time linear systems and others notions for positive linear systems. The considerations can be also extended for 2D linear systems [5].

**Acknowledgment.** The work was supported by the State Committee for Scientific Research of Poland under grant No. 3 T11A 006 27.

## References

1. L. Benvenuti and L. Farina, A tutorial on the positive realization problem, *IEEE Trans. Autom. Control*, vol. 49, No 5, (2004), pp. 651-664.
2. M. Busłowicz, Explicit solution of discrete-delay equations, *Foundations of Control Engineering*, vol. 7, No. 2, (1982), pp. 67-71.
3. M. Busłowicz and T. Kaczorek, Reachability and minimum energy control of positive linear discrete-time systems with one delay, *12<sup>th</sup> Mediterranean Conference on Control and Automation*, June 6-9, (2004), Kauadasi, Izmir, Turkey.
4. L. Farina and S. Rinaldi, *Positive Linear Systems; Theory and Applications*, J. Wiley, New York, (2000).
5. T. Kaczorek, *Positive 1D and 2D Systems*, Springer-Verlag, London (2002).
6. T. Kaczorek, Some recent developments in positive systems, *Proc. 7<sup>th</sup> Conference of Dynamical Systems Theory and Applications*, Łódź (2003), pp. 25-35.
7. T. Kaczorek, Realization problem for positive discrete-time systems with delay, *System Science*, vol. 30, No. 4, (2004), pp. 117-130.
8. T. Kaczorek, Realization problem for positive multivariable discrete-time linear systems with delays in state and inputs. *Proc. Trans. COMP (2005)*, Zakopane, Dec. 5-8, Poland, pp. 1-10.
9. T. Kaczorek, Computation of realizations of discrete-time cone-systems. *Bull. Pol. Acad. Sci. Techn. Sci.*, vol. 54, No. 3, (2006) pp. 347-350.
10. T. Kaczorek Realization problem for positive linear systems with time delay. *Math. Problems in Engineering*, 2005, No. 4, pp. 455-463.
11. T. Kaczorek and M. Busłowicz, Minimal realization for positive multivariable linear systems with delay, *Int. J. Appl. Math. Comput. Sci.*, vol. 14, No 2, (2004), pp. 181-187
12. G. Xie, and L. Wang, Reachability and controllability of positive linear discrete-time systems with time-delays, in L. Benvenuti, A. De Santis and L. Farina (eds): *Positive Systems*, LNCIS 294, Springer-Verlag, Berlin (2003), pp. 377-384.



# Global Stability of Neural Networks with Time-Varying Delays

Yijing Wang and Zhiqiang Zuo\*

School of Electrical Engineering & Automation, Tianjin University,  
Tianjin, 300072, China  
{yjwang, zqzuo}@tju.edu.cn

**Abstract.** This paper deals with the problem of global stability for a class of neural networks with time-varying delays. A new sufficient condition for global stability is proposed by using some slack matrix variables to express the relationship between the system matrices. The restriction on the derivative of the delay function to be less than unit is removed. A numerical example shows that the result obtained in this paper improves the upper bound of the delay over some existing ones.

## 1 Introduction

In the past two decades, the analysis of stability for different classes of neural networks (including Hopfield neural networks, cellular neural networks, Lotka-Volterra neural networks and bi-directional associative memory neural networks) has been extensively studied by many researchers. Neural networks have found many applications in practice, such as pattern recognition, image processing, speed detection of moving objects, optimization problems and so on. A lot of stability conditions have been derived by different methods. On the other hand, time delays are inevitable exist in networks. For example, in hardware implementation of cellular neural networks, time delays are often encountered. It is now well know that time delay can cause instability or oscillation in neural networks. Therefore, study of time delay effects on stability and other dynamics for neural networks has received considerable attention for a long time (see e.g., [1]-[20] and the references therein). These stability results can be classified into two types: that is, delay-independent stability and delay-dependent stability. The former does not include any information on the size of delay while the latter contains such information. Generally speaking, the delay-dependent stability condition is less conservative than the delay-independent one when the size of delay is small. However, it is often assumed in these delay-dependent criteria that the time delay function is continuously differentiable and its derivative does not exceed the unity. As we know, this is a very restrictive condition due to the use of some special Lyapunov-Krasovskii functionals in deriving the stability condition.

Recently, Yang et al. [21] presented delay-dependent global stability and exponential stability criteria for delayed neural networks by choosing an appropriate

---

\* Corresponding author.

Lyapunov-Krasovskii functional. The main advantage of the result in [21] is that the restriction on the derivative of the delay function to be less than unit is removed. As we can see, an inequality is used to obtain the stability condition, which bring additional conservatism in stability analysis. The conservatism of the stability condition of the above paper was reduced by Zuo et al. [22] using some slack matrix variables to express the relationship between the system matrices. In this paper, we derive a new global stability condition by choosing a more general Lyapunov-Krasovskii functional. Neither model transformation nor bounding technique for cross term is involved. It should be noted that our main result is expressed within the framework of linear matrix inequalities (LMI) which can be checked efficiently by the recent interior-point method [23].

The paper is organized as follows. Section 2 gives the model description and some preliminaries. A new delay-dependent stability criterion is presented in Sect. 3. An example is provided to show the reduced conservatism of the proposed condition in Sect. 4. Section 5 concludes the paper.

**Notations:** The following notations are used throughout this paper.  $\Re$  is the set of real numbers. The notation  $X \geq Y$  (respectively,  $X > Y$ ), where  $X$  and  $Y$  are symmetric matrices, means that the matrix  $X - Y$  is positive semi-definite (respectively, positive definite).  $I$  and  $0$  denote the identity matrix and zero matrix with compatible dimensions. In symmetric block matrices, we use an asterisk  $\star$  to denote terms that are induced by symmetry.

## 2 Problem Statement and Preliminaries

The dynamic behavior of a neural network with time-varying delay can be described as

$$\dot{u}(t) = -C u(t) + A g(u(t)) + B g(u(t - \tau(t))) + b \tag{1}$$

or

$$\dot{u}_i(t) = -c_i u_i(t) + \sum_{j=1}^n a_{ij} g_j(u_j(t)) + \sum_{j=1}^n b_{ij} g_j(u_j(t - \tau(t))) + b_i, \quad i = 1, 2, \dots, n \tag{2}$$

where

$$u(t) = [u_1(t) \quad u_2(t) \quad \dots \quad u_n(t)]^T$$

is the neuron state.

$$g(u(t)) = [g_1(u_1(t)) \quad g_2(u_2(t)) \quad \dots \quad g_n(u_n(t))]^T$$

$$g(u(t - h)) = [g_1(u_1(t - h)) \quad g_2(u_2(t - h)) \quad \dots \quad g_n(u_n(t - h))]^T$$

are the activation functions.

$$b = [b_1 \quad b_2 \quad \dots \quad b_n]^T$$

is the constant external input.  $C = \text{diag}\{c_i\} > 0$  is a positive diagonal matrix.  $\tau(t)$  is a continuous function describing the time-varying transmission delays in the network. We assume that

$$0 \leq \tau(t) \leq h, \quad \forall t \geq 0$$

$A$  and  $B$  are interconnection matrices representing the weight coefficients of the neurons.

Throughout this paper, we assume that  $g_j(\cdot)$ ,  $j = 1, 2, \dots, n$  is bounded and satisfy the following condition

$$0 \leq \frac{g_j(\zeta_1) - g_j(\zeta_2)}{\zeta_1 - \zeta_2} \leq k_j \quad j = 1, 2, \dots, n \tag{3}$$

for all  $\zeta_1, \zeta_2 \in \mathfrak{R}$ ,  $\zeta_1 \neq \zeta_2$ ;  $k_j > 0$ . Here we denote

$$K = \text{diag}\{k_1, k_2, \dots, k_n\} > 0$$

Note that these assumptions guarantee there is at least one equilibrium point for neural network (II).

For convenience, we will make the following transformation to neural network (II)

$$x(t) = u(t) - u^* \tag{4}$$

where  $u^* = [u_1^* \ u_2^* \ \dots \ u_n^*]^T$  is an equilibrium point of (II). Under this transformation, system (II) can be rewritten as

$$\dot{x}(t) = -Cx(t) + Af(x(t)) + Bf(x(t - \tau(t))) \tag{5}$$

where  $f_j(x_j(t)) = g_j(x_j(t) + u_j^*) - g_j(u_j^*)$  and  $f_j(0) = 0$ . It is easy to verify that

$$0 \leq \frac{f_j(x_j)}{x_j} \leq k_j, \quad \forall x_j \neq 0, \quad j = 1, 2, \dots, n \tag{6}$$

By (6), we know that for any scalars  $s_{1j} \geq 0, s_{2j} \geq 0$ ,

$$2 \sum_{j=1}^n s_{1j} f_j(x_j(t)) [k_j x_j(t) - f_j(x_j(t))] \geq 0 \tag{7}$$

$$2 \sum_{j=1}^n s_{2j} f_j(x_j(t - \tau(t))) [k_j x_j(t - \tau(t)) - f_j(x_j(t - \tau(t)))] \geq 0 \tag{8}$$

which can be rewritten as

$$2f^T(x(t))S_1Kx(t) - 2f^T(x(t))S_1f(x(t)) \geq 0 \tag{9}$$

$$2f^T(x(t - \tau(t)))S_2Kx(t - \tau(t)) - 2f^T(x(t - \tau(t)))S_2f(x(t - \tau(t))) \geq 0 \tag{10}$$

where

$$S_1 = \text{diag}\{s_{11}, s_{12}, \dots, s_{1n}\} \geq 0$$

$$S_2 = \text{diag}\{s_{21}, s_{22}, \dots, s_{2n}\} \geq 0$$

In order to obtain our main results in this paper, the following lemma is needed.

**Lemma 1.** By (5), for any matrices  $N_i, i = 1, 2, \dots, 5$  with appropriate dimension, we have

$$2 [x^T(t)N_1 + x^T(t - \tau(t))N_2 + f^T(x(t))N_3 + f^T(x(t - \tau(t)))N_4 + \dot{x}^T(t)N_5] \times [\dot{x}(t) + Cx(t) - Af(x(t)) - Bf(x(t - \tau(t)))] = 0 \tag{11}$$

### 3 Main Results

Now we are in the position to derive a new global asymptotic stability criterion for delayed neural network (5).

**Theorem 1.** The neural network with time-varying delay (5) is globally asymptotically stable if there exist matrices  $P > 0$ , three diagonal matrices  $D > 0, S_1 \geq 0, S_2 \geq 0$  and appropriate dimensional matrices  $N_i (i = 1, \dots, 5), X_{ij} (i, j = 1, 2, 3, 4, 5, 6)$  such that the following conditions hold

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} & \Sigma_{14} & \Sigma_{15} \\ * & \Sigma_{22} & \Sigma_{23} & \Sigma_{24} & \Sigma_{25} \\ * & * & \Sigma_{33} & \Sigma_{34} & \Sigma_{35} \\ * & * & * & \Sigma_{44} & \Sigma_{45} \\ * & * & * & * & \Sigma_{55} \end{bmatrix} < 0 \tag{12}$$

$$X = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ * & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ * & * & X_{33} & X_{34} & X_{35} & X_{36} \\ * & * & * & X_{44} & X_{45} & X_{46} \\ * & * & * & * & X_{55} & X_{56} \\ * & * & * & * & * & X_{66} \end{bmatrix} > 0 \tag{13}$$

where

$$\begin{aligned} \Sigma_{11} &= N_1C + CN_1^T + hX_{11} + X_{16} + X_{16}^T \\ \Sigma_{12} &= CN_2^T + hX_{12} - X_{16} + X_{26}^T \\ \Sigma_{13} &= CN_3^T - N_1A + hX_{13} + X_{36}^T + KS_1 \\ \Sigma_{14} &= CN_4^T - N_1B + hX_{14} + X_{46}^T \\ \Sigma_{15} &= CN_5^T + N_1 + P + hX_{15} + X_{56}^T \\ \Sigma_{22} &= hX_{22} - X_{26} - X_{26}^T \\ \Sigma_{23} &= -N_2A + hX_{23} - X_{36}^T \\ \Sigma_{24} &= -N_2B + hX_{24} - X_{46}^T + KS_2 \\ \Sigma_{25} &= N_2 + hX_{25} - X_{56}^T \\ \Sigma_{33} &= -N_3A - A^TN_3^T + hX_{33} - 2S_1 \\ \Sigma_{34} &= -A^TN_4^T - N_3B + hX_{34} \end{aligned}$$

$$\begin{aligned} \Sigma_{35} &= -A^T N_5^T + N_3 + D + hX_{35} \\ \Sigma_{44} &= -N_4 B - B^T N_4^T + hX_{44} - 2S_2 \\ \Sigma_{45} &= -B^T N_5^T + N_4 + hX_{45} \\ \Sigma_{55} &= N_5 + N_5^T + hX_{55} \end{aligned}$$

*Proof.* Consider the following Lyapunov-Krasovskii functional

$$V(x_t) = V_1(x_t) + V_2(x_t) + V_3(x_t) + V_4(x_t) \tag{14}$$

where

$$\begin{aligned} V_1(x_t) &= x^T(t)Px(t) \\ V_2(x_t) &= 2 \sum_{i=1}^n d_i \int_0^{x_i(t)} f_i(s)ds \\ V_3(x_t) &= \int_{t-h}^t \int_{\sigma}^t \dot{x}^T(s)X_{66}\dot{x}(s) dsd\sigma \\ V_4(x_t) &= \int_0^t \int_{\sigma-\tau(\sigma)}^{\sigma} \psi^T X \psi ds d\sigma \end{aligned}$$

$$\psi^T = [x^T(\sigma) \quad x^T(\sigma - \tau(\sigma)) \quad f^T(x(\sigma)) \quad f^T(x(\sigma - \tau(\sigma))) \quad \dot{x}^T(\sigma) \quad \dot{x}^T(s)]$$

and  $X$  is defined in (13).

By (9), (10) and (11), the time derivative of  $V(x_t)$  along the trajectory of neural network (1) is

$$\begin{aligned} \dot{V}(x_t) &= 2x^T(t)P\dot{x}(t) + 2f^T(x(t))D\dot{x}(t) + h\dot{x}^T(t)X_{66}\dot{x}(t) - \int_{t-h}^t \dot{x}^T(s)X_{66}\dot{x}(s)ds \\ &\quad + \tau(t) \xi^T(t)\hat{X}_5\xi(t) + 2\xi^T(t) \begin{bmatrix} X_{16} \\ X_{26} \\ X_{36} \\ X_{46} \\ X_{56} \end{bmatrix} (x(t) - x(t - \tau(t))) \\ &\quad + \int_{t-\tau(t)}^t \dot{x}^T(s)X_{66}\dot{x}(s)ds \\ &\leq 2x^T(t)P\dot{x}(t) + 2f^T(x(t))D\dot{x}(t) + h\dot{x}^T(t)X_{66}\dot{x}(t) \\ &\quad + \tau(t) \xi^T(t)\hat{X}_5\xi(t) + 2\xi^T(t) \begin{bmatrix} X_{16} \\ X_{26} \\ X_{36} \\ X_{46} \\ X_{56} \end{bmatrix} (x(t) - x(t - \tau(t))) \\ &\quad + 2f^T(x(t))S_1Kx(t) - 2f^T(x(t))S_1f(x(t)) \\ &\quad + 2f^T(x(t - \tau(t)))S_2Kx(t - \tau(t)) - 2f^T(x(t - \tau(t)))S_2f(x(t - \tau(t))) \\ &\quad + 2[x^T(t)N_1 + x^T(t - \tau(t))N_2 + f^T(x(t))N_3 + f^T(x(t - \tau(t)))N_4 \\ &\quad \quad + \dot{x}^T(t)N_5] \times [\dot{x}(t) + Cx(t) - Af(x(t)) - Bf(x(t - \tau(t)))] \\ &= \xi^T(t) \Sigma \xi(t) \end{aligned} \tag{15}$$

where

$$\xi^T(t) = [x^T(t) \quad x^T(t - \tau(t)) \quad f^T(x(t)) \quad f^T(x(t - \tau(t))) \quad \dot{x}^T(t)]$$

$$\hat{X}_5 = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} \\ \star & X_{22} & X_{23} & X_{24} & X_{25} \\ \star & \star & X_{33} & X_{34} & X_{35} \\ \star & \star & \star & X_{44} & X_{45} \\ \star & \star & \star & \star & X_{55} \end{bmatrix}$$

and  $\Sigma$  is defined in (12). If  $\Sigma < 0$ , then  $\dot{V}(x_t) < 0$  for any  $\xi(t) \neq 0$ , which guarantees the asymptotically stable for delayed neural network (5). This completes the proof.

*Remark 1.* Theorem 1 presented a novel global asymptotic stability condition for delayed neural network by using a more general Lyapunov-Krasovskii functional based on the result of [22]. It should be noted that our main result is expressed within the framework of linear matrix inequalities which can be easily computed by the interior-point method [23].

*Remark 2.* We can easily obtain the exponential stability condition by the method proposed in [21]. That is, let  $z(t) = e^{kt}x(t)$  where  $k$  is a positive constant which provides an estimate of the exponential decay rate of system. Then system (5) can be transformed to the one which is suitable for exponential stability analysis.

*Remark 3.* The method proposed in this paper can be used to deal with the problem of robust stability for uncertain neural networks with time-varying delays. The linear fractional form studied in [24] is more general than the norm-bounded ones [14].

## 4 Example

In this section, we give an example to show the reduced conservatism of the proposed condition. Let us consider the same delayed neural network (5) which was studied in [21] and [22]

$$C = \begin{bmatrix} 0.7 & 0 \\ 0 & 0.7 \end{bmatrix}, \quad A = \begin{bmatrix} -0.3 & 0.3 \\ 0.1 & -0.1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.1 & 0.1 \\ 0.3 & 0.3 \end{bmatrix}$$

and

$$f_1(x) = f_2(x) = 0.5(|x + 1| - |x - 1|)$$

Using the Matlab LMI-Toolbox to solve this problem, we found that LMI (12) and (13) is feasible for any arbitrarily large  $h$  by Theorem 1 in this paper (as long as the numerical computation is reliable). Compared with the result of  $h = 0.29165$  in [21] and  $h = 2.1570$  in [22], it is obvious that our method

provides a less conservative result. For example, if we choose  $h = 30$ , we can obtain the following results by Theorem 1

$$\begin{aligned}
 S_1 &= \begin{bmatrix} 9.5901 & 0 \\ 0 & 4.9156 \end{bmatrix}, & S_2 &= \begin{bmatrix} 5.1055 & 0 \\ 0 & 3.5593 \end{bmatrix} \\
 P &= \begin{bmatrix} 38.0973 & -13.9096 \\ -13.9096 & 16.5997 \end{bmatrix}, & D &= \begin{bmatrix} 18.5987 & 0 \\ 0 & 6.3502 \end{bmatrix} \\
 X_{11} &= \begin{bmatrix} 7.7576 & -0.1209 \\ -0.1209 & 7.4421 \end{bmatrix}, & X_{12} &= \begin{bmatrix} -0.2844 & -0.1119 \\ -0.1177 & -0.2686 \end{bmatrix} \\
 X_{13} &= \begin{bmatrix} 3.0428 & -3.0311 \\ -1.0300 & 1.0153 \end{bmatrix}, & X_{14} &= \begin{bmatrix} -0.9684 & -0.9737 \\ -2.9772 & -2.9852 \end{bmatrix} \\
 X_{15} &= \begin{bmatrix} 9.5642 & 0.1094 \\ 0.1020 & 9.8082 \end{bmatrix}, & X_{16} &= \begin{bmatrix} -4.9605 & -2.0165 \\ -2.0570 & -4.8510 \end{bmatrix} \\
 X_{22} &= \begin{bmatrix} 0.3781 & 0.1532 \\ 0.1532 & 0.3616 \end{bmatrix}, & X_{23} &= \begin{bmatrix} 0.0423 & -0.0375 \\ 0.0048 & -0.0018 \end{bmatrix} \\
 X_{24} &= \begin{bmatrix} -0.1265 & -0.0294 \\ -0.0487 & -0.1129 \end{bmatrix}, & X_{25} &= \begin{bmatrix} 0.1595 & 0.0436 \\ 0.0602 & 0.1402 \end{bmatrix} \\
 X_{26} &= \begin{bmatrix} 7.1759 & 2.8486 \\ 2.8518 & 6.8907 \end{bmatrix}, & X_{33} &= \begin{bmatrix} 1.9109 & -1.4597 \\ -1.4597 & 1.5694 \end{bmatrix} \\
 X_{34} &= \begin{bmatrix} -0.0398 & -0.0376 \\ -0.0253 & -0.0263 \end{bmatrix}, & X_{35} &= \begin{bmatrix} 4.1415 & -1.3953 \\ -4.2651 & 1.4010 \end{bmatrix} \\
 X_{36} &= \begin{bmatrix} 0.7890 & 0.0693 \\ -0.7876 & -0.0721 \end{bmatrix}, & X_{44} &= \begin{bmatrix} 1.6063 & 1.4197 \\ 1.4197 & 1.5524 \end{bmatrix} \\
 X_{45} &= \begin{bmatrix} -1.4453 & -4.3000 \\ -1.4479 & -4.3102 \end{bmatrix}, & X_{46} &= \begin{bmatrix} -0.6192 & -0.9620 \\ -0.6444 & -0.9561 \end{bmatrix} \\
 X_{55} &= \begin{bmatrix} 14.7951 & -0.0987 \\ -0.0987 & 14.5729 \end{bmatrix}, & X_{56} &= \begin{bmatrix} 2.8730 & 1.1773 \\ 1.2184 & 2.8344 \end{bmatrix} \\
 X_{66} &= \begin{bmatrix} 218.0436 & 87.3772 \\ 87.3772 & 209.2223 \end{bmatrix}, & N_1 &= 10^3 \times \begin{bmatrix} -0.1702 & -1.4303 \\ 1.4438 & -0.1613 \end{bmatrix} \\
 N_2 &= \begin{bmatrix} -2.6702 & -0.2736 \\ -0.9706 & -2.2223 \end{bmatrix}, & N_3 &= \begin{bmatrix} -280.1523 & -596.3896 \\ 266.8339 & 590.0440 \end{bmatrix} \\
 N_4 &= \begin{bmatrix} -593.7328 & 269.9261 \\ -593.5394 & 270.4726 \end{bmatrix}, & N_5 &= 10^3 \times \begin{bmatrix} -0.2316 & -2.0506 \\ 2.0548 & -0.2272 \end{bmatrix}
 \end{aligned}$$

## 5 Conclusion

An improved delay-dependent global stability criterion for a class of delayed neural networks has been derived in this paper based on a novel Lyapunov-Krasovskii functional. This condition is expressed within the framework of linear matrix inequalities. An example has been provided to demonstrate the less conservatism of the proposed result.

**Acknowledgments.** This work is supported by National Natural Science Foundation of China (No. 60504011) and (No. 60504012).

## References

1. Arik, S.: An analysis of global asymptotic stability of delayed cellular neural networks. *IEEE Trans. Neural Networks.* **13** (2002) 1239–1242
2. Arik, S.: Global robust stability of delayed neural networks. *IEEE Trans. Circuits Syst. I.* **50** (2003) 156–160
3. Cao, J.: A set of stability criteria for delayed cellular neural networks. *IEEE Trans. Circuits Syst. I.* **48** (2001) 494–498
4. Cao, J., Huang, D.S., Qu, Y.Z.: Global robust stability of delayed recurrent neural networks. *Chaos, Solitons and Fractals.* **23** (2005) 221–229
5. Chen, A., Cao, J., Huang, L.: An estimation of upperbound of delays for global asymptotic stability of delayed Hopfield neural networks. *IEEE Tran. Circuit Syst. I.* **49** (2002) 1028–1032
6. Chu, T. G.: Necessary and sufficient condition for absolute stability of normal neural networks. *Neural Networks.* **16** (2003) 1223–1227
7. Chu, T. G.: A decomposition approach to analysis of competitive/cooperative neural networks with delay. *Physics Letters A.* **312** (2003) 339–347
8. Chu, T. G.: An exponential convergence estimate for analog neural networks with delay. *Physics Letters A.* **283** (2001) 113–118
9. He, Y., Wang, Q.G., Wu, M.: LMI-based stability criteria for neural networks with multiple time-varying delays, *Physica D*, **212** (2005) 126–136
10. Li, C.D., Liao, X.F., Zhang, R.: Global robust asymptotical stability of multi-delayed interval neural networks: an LMI approach. *Physics Letters A.* **328** (2004) 452–462
11. Liang, J.L., Cao, J.: A based-on LMI stability criterion for new criterion for delayed recurrent neural networks. *Chaos, Solitons and Fractals.* **28** (2006) 154–160
12. Liao, X.F., Wong, K.W., Wu, Z.F., Chen, G.: Novel robust stability criterion for interval-delayed Hopfield neural networks. *IEEE Trans. Circuits Syst. I.* **48** (2001) 1355–1359
13. Liao, X.F., Chen, G., Sanchez, E.N., Delay-dependent exponential stability analysis of delayed neural networks: an LMI approach. *Neural Networks.* **15** (2002) 855–866
14. Singh, V.: Robust stability of cellular neural networks with delay: linear matrix inequality approach. *IEE Proc.-Control Theory Appl.* **151** (2004) 125–129
15. Singh, V.: Global robust stability of delayed neural networks: an LMI approach. *IEEE Trans. Circuits Syst. II.* **52** (2005) 33–36
16. Xu, S., Lam, J., Ho, D.W.C., Zou, Y.: Novel global asymptotic stability criteria for delayed cellular neural networks. *IEEE Tran. Circuit Syst. II.* **52** (2005) 349–353



17. Ye, H., Michel, A.N., Wang, K.: Global stability and local stability of Hopfield neural networks with delays. *Physical Review E*. **59** (1994) 4206–4213
18. Zuo, Z.Q., Wang, Y.J.: Novel delay-dependent exponential stability analysis for a class of delayed neural networks. *Lecture Notes in Computer Science*. Springer Verlag. **4113** (2006) 216–226
19. Zhang, H., Li, C.G., Liao, X.F.: A note on the robust stability of neural networks with time delay. *Chaos, Solitons and Fractals*. **25** (2005) 357–360
20. Zhang, Q., Wei, X., Xu, J.: Global asymptotic stability of Hopfield neural networks with transmission delays. *Physics Letters A*. **318** (2003) 399–405
21. Yang, H.F., Chu, T., Zhang, C.: Exponential stability of neural networks with variable delays via LMI approach. *Chaos, Solitons and Fractals*. **30** (2006) 133–139
22. Zuo, Z.Q., Wang, Y.J.: Global asymptotic stability analysis for neural networks with time-varying delays, in 45th IEEE Conference on Decision and Control, San Diego, USA, 2006.
23. Boyd, S., Ghaoui, L. El., Feron, E., Balakrishnan, V.: *Linear matrix inequalities in systems and control theory*, Philadelphia, SIAM, (1994)
24. Zuo, Z.Q., Wang, Y.J.: Relaxed LMI condition for output feedback guaranteed cost control of uncertain discrete-time systems. *Journal of Optimization Theory and Applications*. **127** (2005) 207–217

# A Sensorless Initial Rotor Position Sensing Using Neural Network for Direct Torque Controlled Permanent Magnet Synchronous Motor Drive

Mehmet Zeki Bilgin

Kocaeli University, Department of Electrical Engineering, Vinsan Kampusu,  
41300, Kocaeli, Turkey  
Bilgin@kou.edu.tr  
<http://www.kou.edu.tr>

**Abstract.** This paper presents a method to determine the initial rotor position for Direct Torque Controlled (DTC) Permanent Magnet Synchronous Motor (PMSM) using Artificial Neural Network (ANN). The inductance variation is a function of the rotor position and stator current for PMSM. A high frequency and low magnitude voltage is applied to the stator windings and examined the effects the stator currents by using ANN for initial rotor position detection.

## 1 Introduction

Recent advances in power semiconductor devices, microprocessor, converter design technology and control theory have enabled ac servo drives to satisfy the high performance requirements in many industrial applications. The Permanent Magnet Synchronous Motors (PMSM) are used in many applications that require rapid torque response and high-performance operation such as robotics, vehicle propulsion, heat pumps, actuators, machine tools and ship propulsion. In these applications, the PMSM drive systems are required to position or velocity feedback.

The Direct Torque Control (DTC) is recently developed and adopted to PMSM drive systems [1]. If the initial rotor position is known at  $t=0$ , the motor is successfully started. But if the initial position is not known, the motor may be temporarily rotated into wrong direction.

In most applications, there is an optical shaft position encoder or resolver for position feedback signal. There are several disadvantages of resolver or encoder such as higher number of connection between motor and its driver, additional cost, susceptibility to noise and vibrations, extra space, volume and weight on the motor. All these disadvantages restricted the performance of the PMSM. Moreover, the use of cheap incremental encoders does not allow the knowledge of the initial rotor position.

An initial rotor position estimator without any mechanical shaft sensor is desirable for the DTC scheme of PMSM. Some techniques were proposed in literature. The techniques are mostly based on the Kalman filters [2] or state observer [3], magnetic saliency [4], high frequency voltage signal injection [5], [6] and back emf measurement. Some of these schemes are not suitable for initial rotor position detection such as back emf measurement. The back emf is zero at standstill.

In this paper, a method to detect the rotor position at start-up of PMSM is based on the magnetic saturation principle in reference [7]. The motor equations are related to the windings inductances, currents, voltages, the rotor position and the motor parameters. The initial rotor position may be derived from lookup table that is calculated motor windings inductances variation. In this work, the rotor position is estimated by off-line trained Neural Network (NN). The three phase high frequency low magnitude test voltage signals are applied to stator windings and the phase currents data for many rotor positions are measured and stored. The NN is trained by off-line with stored data. At the control mode, the same test signals are injected into the motor, the phase currents are measured and these currents are applied to NN. The NN derived the initial rotor position. The estimation performance is increased and position estimation error is minimised using the NN estimator.

## 2 System Description

Figure 1 shows the simplified block diagram of a PMSM control system investigated in this work.

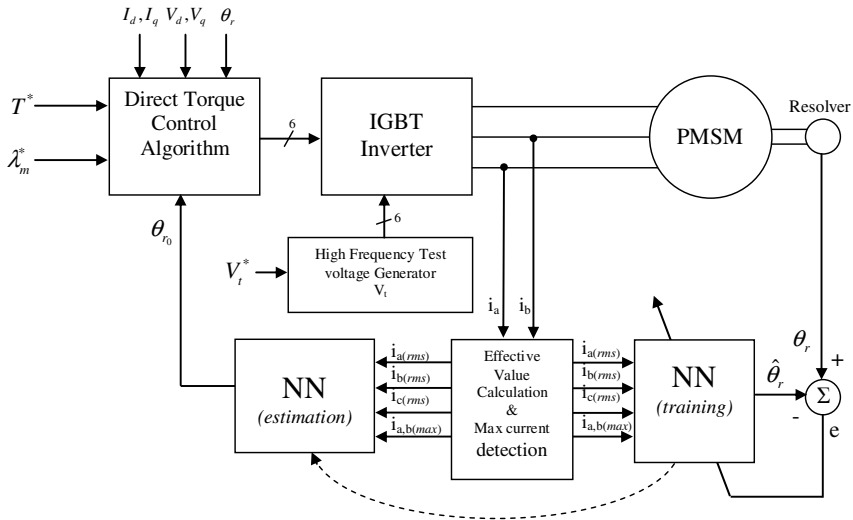


Fig. 1. The proposed initial rotor position estimation structure

The proposed initial rotor position scheme consists of three main stages. First, the high frequency and low magnitude test phase voltage are injected to phase windings. Immediately after, the phase currents are measured and stored. Second, the effective value of the stored currents are calculated and applied to NN for learning. Finally, the initial rotor position is predicted by NN. The PMSM is driven by

IGBT (Insulated-Gate Bipolar Transistors) inverter. The inverter switches are controlled by Direct Torque Control (DTC) algorithm. The reference signals are torque and flux. The feedback signals are stator phase voltages, currents and rotor position. The initial rotor position is required at only start-up for DTC. Therefore, there is no function of the NN predictor for the normal operation mode. The test and control voltages are generated from same IGBT inverter. At the test mode, the inverter produces low magnitude, high frequency voltages using  $V_t^*$  reference command. These voltages can not cause any movement at the rotor. The initial rotor position is estimated by NN. At the control mode, the inverter is controlled by the DTC algorithm and the inverter produces the control voltage of PMSM with the given reference signals.

The rotor is rotated to any position and this point data is stored. The low magnitude and high frequency voltages are applied to the motor and the 3 phase currents are measured and stored. The same procedure is repeated for each five degree from 0 to 180°. The 36 data set is prepared for this work and the NN is trained with this data. Before the normal drive mode, the same test voltage is applied to PMSM and the currents responses are measured. These current's effective and maximum values are calculated and applied to the trained NN. The initial rotor position is produced at the out of the NN. The DTC algorithm uses this initial rotor position data.

## 2.1 The Direct Control of PMSM

The Direct Torque Control (DTC) scheme is established to calculation of the stator phase voltage according to reference torque and flux. The current control algorithm is not used. Thus, The DTC needs only the stator resistance value. The stator flux vector is predicted. From the stator flux equation [7];

$$\frac{d\phi_s}{dt} = V_s - i_s r_s \quad (1)$$

$$\phi_s = v_s t - r_s \int i_s dt + \phi_s|_{t=0} \quad (2)$$

The rotor position of the PMSM at  $t=0$  must be known. It is shown from (1) and (2). The DTC scheme is clearly described in reference [1]. Therefore, it is not repeated here. Only the proposed method is explained for calculation of  $\Phi_s|_{t=0}$ .

## 2.2 The Mathematical Model of PMSM

For the simulation work, the PMSM mathematical model is developed. It assumed that the machine parameters are not constant and the inverter switches are ideal. The inverter voltage harmonics and operation temperature are neglected. The structure of PMSM and equivalent circuit of stator winding are shown in Fig. 2.

The PMSM has three phases, star connected four poles, 3 HP, 5000 round/minute and the stator windings are equally and sinusoidal distributed. The phase voltage equations of PMSM are written as follows according to Fig. 2.b.

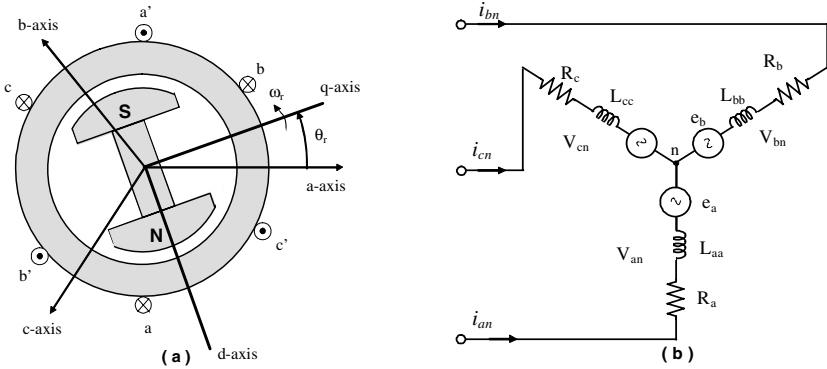


Fig. 2. (a) The structure of the two poles PMSM, (b) The stator windings

$$V_{an} = r_s i_{an} + L_{ss} \frac{di_{an}}{dt} + \omega_r \lambda_m \cos(\theta_r) \tag{3}$$

$$V_{bn} = r_s i_{bn} + L_{ss} \frac{di_{bn}}{dt} + \omega_r \lambda_m \cos\left(\theta_r - \frac{2\pi}{3}\right) \tag{4}$$

$$V_{cn} = r_s i_{cn} + L_{ss} \frac{di_{cn}}{dt} + \omega_r \lambda_m \cos\left(\theta_r + \frac{2\pi}{3}\right) \tag{5}$$

where  $r_s$ ,  $L_{ss}$ ,  $\theta_r$ ,  $\omega_r$ , and  $\lambda_m$  are per-phase stator resistance, stator self inductance, the position of the rotor, angular shaft speed and the flux linkage due to permanent magnet ,respectively. The phase voltage is transformed into d-q axis;

$$V_{qs}^r = V_t \sin(\theta_h - \theta_r) = (r_s + pL_q) i_{qs}^r + \omega_r L_d i_{ds}^r + \omega_r \lambda_m \tag{6}$$

$$V_{ds}^r = V_t \cos(\theta_h - \theta_r) = (r_s + pL_q) i_{ds}^r - \omega_r L_q i_{qs}^r \tag{7}$$

$$\theta_r = \int_0^t \omega_r dt + \theta_{r0} \quad \theta_h = \int_0^t \omega_h dt + \theta_{h0} \tag{8}$$

The electrical torque is;

$$T_e = \left(\frac{3}{2}\right)\left(\frac{P}{2}\right)\left(\lambda_m i_{qs}^r + (L_d - L_q) i_{qs}^r i_{ds}^r\right) \tag{9}$$

$$T_e = J\left(\frac{2}{P}\right) \cdot \frac{d\omega_r}{dt} + B_m\left(\frac{2}{P}\right) \cdot \omega_r + T_L \tag{10}$$

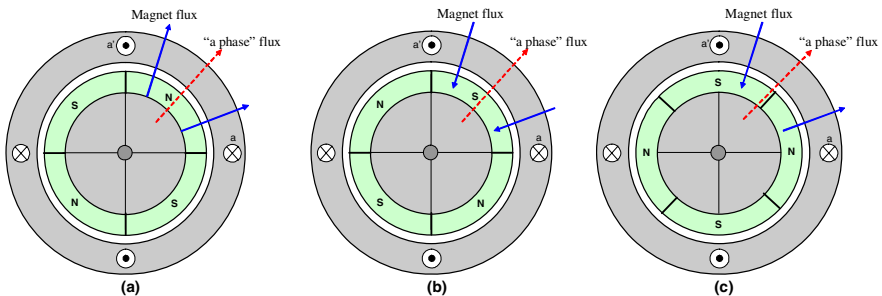
where  $J$ ,  $T_L$ ,  $B_m$  and  $P$  are inertia, load torque, friction constant and pole number of PMSM, respectively.

### 3 Proposed Method

The rotor does not move when the motor’s stator windings are supplied with the low magnitude and high frequency voltages. It is shown from the developed model that the rotor is rotated by  $\pm 8.6e-3$  degree when the windings are supplied 35 Volt, 350 Hz voltages. This displacement may be neglected. Thus, the current of each phase is related to standstill rotor position.

For example, when the stator inductance variation of the four pole surface mounted PMSM is studied for probable 3 rotor positions, the three main structures may be driven as shown in Fig. 3. The shapes are drawn for only “phase a” and easily also drawn for the others. In the Figure 3.a, the north (N) pole is aligned with the coil and the winding’s flux and magnet flux is in the same direction. The winding’s current increases the total flux, increases saturation and decreases inductance. When the rotor is rotated about 180 degree, the S pole is aligned with the “a phase” windings axis. This position is shown in Fig. 3.b. The current in the winding decreases the flux and increases the inductance. Figure 3.c shows the rotor rotated to the position where the intersection of south and north poles is aligned with the coil. At this state, the inductance of winding is different from the others.

Figure 4.a shows the stator phase inductance variation produced by only rotor magnet and Figure 4.b shows stator phase inductance variation produced by rotor magnet



**Fig. 3.** The possible position between magnet flux and stator windings flux

and stator currents [8]. When the stator iron is saturated, the inductance of the coil is reduced. Thus, the phase currents are different for different rotor position. The rotor position may be estimated with using these currents.

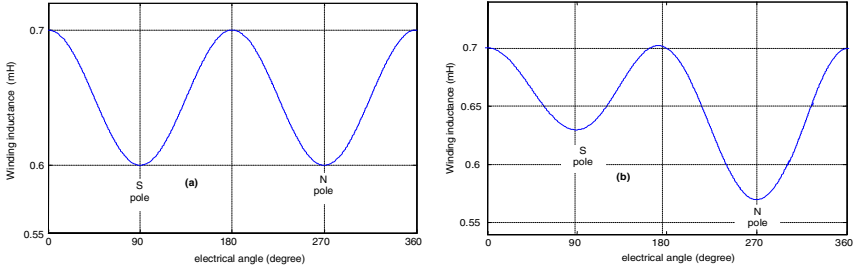


Fig. 4. The stator inductance variation with; (a) only rotor flux, (b) stator currents and rotor flux

The phase currents are shown in Fig. 5 for different rotor positions when the windings are supplied by 35 Volt, 350 Hz voltages. The effective value of the current is calculated using stored data. This effective values and maximum values of phase currents are inputs and its rotor position is target vector for NN (Fig. 6) The NN structure is 3:12:1, tanh:tanh:purelin and Feed Forward (FF). The NN is trained with Levenberg-Marquardt algorithm.

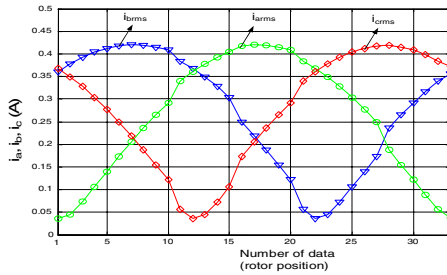


Fig. 5. The variations of the effective value of the three phase current versus different rotor positions

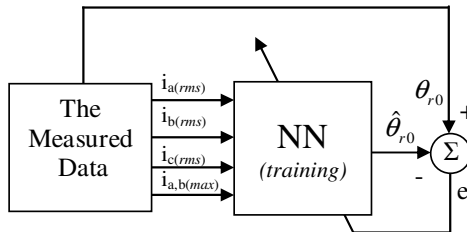
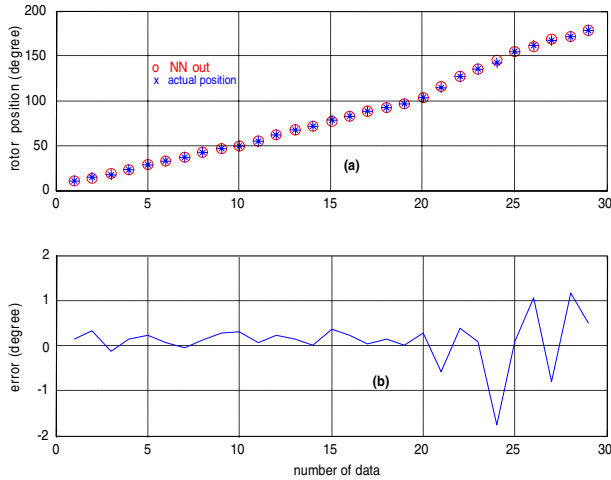


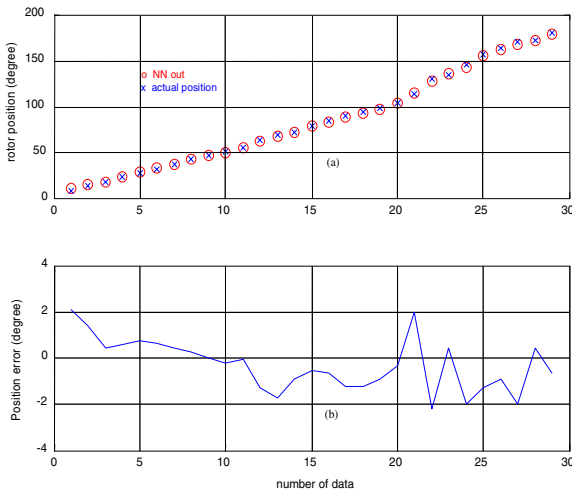
Fig. 6. The training structure of Neural Network

After the training, the NN structure and weight is saved. When the controller needs the initial rotor position any time, the windings are supplied by 35 Volt, 350 Hz volt-ages. The phase currents are measured, the effective values of currents are calculated, the maximum values of phase currents are obtained and all of these are applied to NN. The initial rotor position is produced to out of NN. Figure 7.a is a plot of the actual and estimated initial rotor positions for 29 test data. Figure 7.b shows the estimation errors.

It is shown that the maximum estimation error is  $1,8^\circ$ . Figure 8.a shows the actual and estimated initial rotor positions when the stator resistance is increased a twenty-five percent. The maximum estimation error is  $2,1^\circ$  and shown in Fig. 8.b.



**Fig. 7.** The rotor position and position error with nominal parameters



**Fig. 8.** The rotor position and position error with increasing the stator resistance up to %125



Figure 9 shows initial rotor position errors when q-axis inductance of the stator windings is decreased to twenty-five percent. Nevertheless, the maximum estimation error is  $2,2^\circ$ .

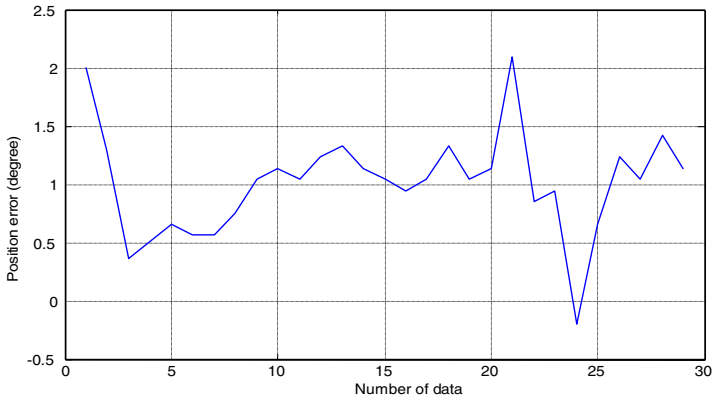


Fig. 9. The rotor position error with decreasing the stator q-axis inductance (%75  $L_q$ )

## 4 Finding of Pole Direction

All of the above work estimates only initial rotor position angle. Also the rotor pole direction may be estimated. When the Fig. 3 is examined, it is shown that the current's magnitude depended on pole direction. For the finding of pole direction, the low magnitude square-wave pulse is applied to stator phase windings and the current is measured and stored. After same pulse applied to winding as reverse direction and the current is measured and stored too. The pole direction is determined by comparing these two currents [9], [10].

## 5 Conclusions

The focus of this paper is to estimate the initial rotor position angle of PMSM at standstill. The new estimation approach based on NN is described. In sensorless DTC scheme, drive systems are not equipped with position resolver or encoder. The use of proposed method allows the initial rotor position to estimate. The NN is trained off-line according to the phase currents with given rotor position. The results show that NN estimator is better than the conventional estimator for initial rotor estimation. The position error is 1,8 degree with the nominal motor parameters and it is 2,1 degree if the stator resistance is changed to 1,25 times the nominal value. When the stator q-axis inductance ( $L_q$ ) is changed 0,75 times of the nominal value, the maximum position error is 2,2 degree. These results show that the rotor does not rotate into wrong direction. The different parameters may be used instead of effective and maximum value of the stator currents and the similar result may be occurred. Because

the NN's training sets are related to motor parameter and test voltage, the NN must be trained again for other motors.

## References

1. Zhong L., Rahman M.F., Hu W.Y., Lim K.W.: Analysis of Direct Torque Control in Permanent Magnet Synchronous Motor Drives, *IEEE Trans. on Power Electronics*, Vol.12, No 3, (May 1997)
2. Dhaouadi R., Mohan N., Norum L.: Design and Implementation of Extended Kalman Filter for the state Estimation of a Permanent Synchronous Motor, *IEEE Trans. on Power Electronics*, Vol.6, pp-491-497, (1991)
3. Low T.S., Lee T.H., Chang K.T.: A non-linear speed observer for Permanent Magnet Synchronous Motor, *IEEE Trans. on Ind. Electronics*, Vol.40, pp-307-316, (1991)
4. Jansen P.L., Lorenz R.D.: Transducerless Position and Velocity Estimation in Induction and salient AC machines, *IEEE Industry Application*, Vol.31, no 2, pp-240-247,(1995)
5. Consili A., Scarcella G., Testa A., Sensorless Control of AC Motors at Zero Speed, *IEEE ISIE*, Bled , Slovenia, (1999)
6. Noguchi T., Yamada K., Kondo S., Takahashi I.: Initial Rotor Position Estimation Method of Sensorless PM Motor with No Sensitivity to Armature Resistance, *IEEE Tran. on Ind. Electronics*, VOL.45, pp-118-125, (1998)
7. Haque M.E., Zhong L. and Rahman M.F.: A Sensorless Initial Rotor Position Estimation Scheme for a Direct Torque Controlled Interior Permanent Magnet Synchronous Motor Drive, *IEEE Trans. on Power Electronics*, Volume: 18 , pp. 1376 - 1383, Nov. (2003)
8. Schmidt P.B., Gasperi M.L., Ray G., Wijenayake A.H.: Initial Rotor Angle Detection of A Non-Salient Pole Permanent Magnet Synchronous Machine, *IEEE Industry Application Society Annual Meeting*, New Orleans, Louisiana, October 5-9, (1997)
9. Haque M.E., Zhong L. and Rahman M.F.: Initial Rotor Position Estimation of Interior Permanent Magnet Synchronous Motor without a Mechanical Sensor, *Journal of Electrical and Electronic Engineering*, Vol. 21, (2001)
10. Parasiliti F., Petrella R., Tursini M.: Initial Rotor Position Estimation Method for PM Motors, *IAS 2000 Annual Meeting and World Conference on Industrial applications of Electrical Energy*, Roma, 8-12 , (2000)

# Postural Control of Two-Stage Inverted Pendulum Using Reinforcement Learning and Self-organizing Map

Jae-kang Lee<sup>1</sup>, Tae-seok Oh<sup>1</sup>, Yun-su Shin<sup>2</sup>, Tae-jun Yoon<sup>3</sup>, and Il-hwan Kim<sup>1</sup>

<sup>1</sup> Department of Electrical and Electronic Engineering, College of Information Technology  
Kangwon National Univ., Chuncheon, Korea  
{margrave, ots, ihkim}@cclab.kangwon.ac.kr

<sup>2</sup> Program of Electronic & Communications Engineering, Department of Electric & Electronic  
Engineering, College of Information Technology  
Kangwon National Univ., Chuncheon, Korea  
bw1709@cclab.kangwon.ac.kr

<sup>3</sup> Department of Mechatronics Engineering, Kangwon National Univ., Chuncheon, Korea  
endmyion@naver.com

**Abstract.** This paper considers reinforcement learning control with the self-organizing map. Reinforcement learning uses the observable states of objective system and signals from interaction of the system and environment as input data. For fast learning in neural network training, it is necessary to reduce learning data. In this paper, we use the self-organizing map to partition the observable states. Partitioning states reduces the number of learning data which is used for training neural networks. And neural dynamic programming design method is used for the controller. For evaluating the designed reinforcement learning controller, a double linked inverted pendulum on the cart system is simulated. The designed controller is composed of serial connection of self-organizing map and two Multi-layer Feed-Forward Neural Networks.

## 1 Introduction

In the case of neural network-based controls that use the supervised learning method, accurate action data is required in advance for the neural network controller's learning, and it is virtually impossible to gain action data for all control object states. As a method capable of compensating for these weaknesses, the reinforcement learning control method was presented, where the learning takes place through interaction between the object and the environment while armed only with such basic information as control target and control range[1]. Reinforcement learning uses observable control object's state and reinforcement signals in the neural network controller's learning. Reinforcement signals are derived from interaction between the control object and the environment, using evaluation of the action the controller took under current states. Evaluation is generally classified into either success or failure, using two scalar values for a simplified expression. And in order to apply this analog signal to the digital control, discretization through sampling is required. Although using these states directly in neural network controller learning allows for precision learning, it remains unsuitable for real applications due to the extended amount of learning

time required [2],[3]. Therefore, it becomes necessary to lower the number of states to within the possible learning range, while maintaining the accuracy level, to accelerate the learning process. In decreasing the number of states, one proposed method calls for using pre-designated values to classify the states and to apply the CMAC (Cerebellar Model Articulation Controller) technique to the states[4],[5]. The underlying problem with this proposed method is that it requires additional information to pre-designated values in order to assign a basis value capable of preserving the distribution characteristics of the states during the classification process. Another proposed method for lowering the number of states involves the use of a self-organizing map (SOM), a neural network variant [6], [7]. SOM preserves distribution characteristics through competitive learning while lowering the number of states [8].

The reinforcement learning controller consists of an evaluation network that evaluates the controller's actions and an action network that assumes actions regarding the given states based on the above evaluation. Learning takes place based on the two data described above as either Temporal Difference (TD) learning or Heuristic Dynamic Programming (HDP). However, the two methods differ in their respective methods of determining action. In the TD learning method the action network learns the action policy and actions are determined based on that very policy [4], [9]. Conversely, in HDP learning the action network does not learn an entire policy, but rather input states and corresponding actions. The action network effectively learns connections for actions per each state, resulting in faster learning rate over the TD learning method.

In this study, for the purpose of accelerated learning, the number of states used in a neural network controller's learning was reduced using a self-organizing map for which its size and classification object states were determined based on basic information that could be gained in advance. Further, an adaptive critic design that bases on neural dynamic programming (NDP), a form of HDP learning, was used to resolve the online learning rate issue that occurs when applying reinforcement learning controllers to actual systems. Lastly, the designed controller was applied to a two-stage inverted pendulum system to verify the controller design's efficacy.

## 2 Controller Composition

The controller consists of action network, evaluation network, self-organizing map that classifies the observed states, and connection from the control object. The self-organizing map and the controller's learning process are described in detail in the following sections.

### 2.1 Self-organizing Map

States observed by the control object system is a digital signal gained through sampling of an analog signal. The number of this digital signal can increase or decrease depending on the sampling period. Slow sampling decreases state data accuracy. The learning process may be accelerated, but the lowered data accuracy results in a lowered learning success rate. Ergo, efficient learning becomes possible if state data with greater accuracy can be gained and the number of data can be reduced while

simultaneously retaining the characteristics of the state data. For this very purpose this study has utilized a self-organizing map (SOM). SOM initializes with each unit positioned randomly on the grid. The units are then rearranged on the grid through competitive learning of learning object data while maintaining the data's characteristics. Here, the rearranged units' weight vectors signify the classified input data. In this study the number of units and the dimension of the map were determined based on the number of observable states and two basic and in advance factors: control range and sampling period.

### 2.2 Neural Dynamic Programming

Objects that the neural network has to learn in Heuristic Dynamic Programming based reinforcement learning is categorized into two. The first category is of objects that need to be learned from the evaluation network, where the evaluation network renews the network's weight so that it approximates the function that represents the evaluation of the action assumed by the controller. The second category is of objects that need to be learned from the action network, where the weight is renewed to select what action to assume under the current states so that the controller action evaluation is maximized. In this study neural dynamic programming (NDP), which bases on action dependent heuristic dynamic programming (ADHDP), a method where the action assumed at the controller is used as evaluation network's input, was adopted. It differs from ADHDP in that it does not use the object system's model. The function that the action network has to learn in NDP is gained through the following equation:

$$J^*(X(t)) = \min_{u(t)} \{ J^*(X(t+1)) + g(X(t), X(t+1)) - U_0 \} \tag{1}$$

Here,  $g(X(t), X(t+1))$  signifies the cost generated by  $u(t)$  at time  $t$ , and  $U_0$  is a heuristic term added to balance the two members. In order to apply  $J(X(t))$  to the evaluation network, the right side of the equal sign from equation (1) must be known beforehand.

Figure 1 is a block diagram that also includes the control object system:

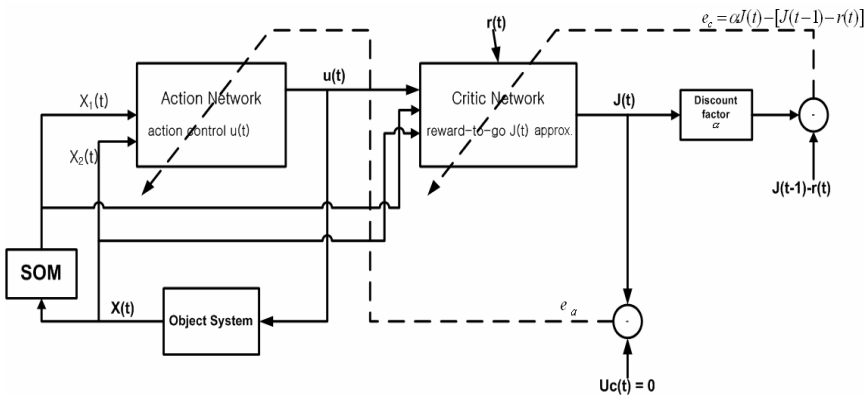


Fig. 1. Control System Block Diagram

### 2.3 Evaluation Network and Action Network Learning

The evaluation network and the action network are both 2-layer feed-forward networks. Specific to this study, a network comprising of a hidden layer and an output layer were used. Transfer functions used in each of the layers are shown below:

$$\text{Hidden layer transfer function : } y = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{2}$$

$$\text{Output layer transfer function : } y = x$$

Inputs of the evaluation network are observable states of control object and  $u$  which is the output of the action network. The action network's formation is identical to the evaluation network, except that it does not include the  $u$  in the input.

$r(t)$ , which is the evaluation network's external reinforcement signal, carries the value "0" when the action assumed by the controller is a success and "-1" if the action is a failure. First, all weights are randomly initialized when learning of the two networks commences. As the current state is observed, the action network assumes an action based on the current weight. If an action that provides a better evaluation for the given state is assumed, it becomes closer to the optimum action sequence gained from Principle of Optimality. In other words, optimum action sequence is one for which the greatest evaluation is given. Therefore, the weight at the action network renews in the direction of optimum action sequence. The evaluation network evaluates the action based on the state, the action, and the reinforcement signal. The evaluation network's output  $J(t)$  from fig.1 is a total of all evaluations to be gained in advance to time  $t$ . In this study the evaluation network's weight is renewed so that  $J(t)$  approximates  $R(t)$ .

$$R(t) = r(t+1) + \alpha r(t+2) + \dots \tag{3}$$

Here,  $R(t)$  represents the evaluation total at the point discounted forward from time  $t$ ,  $\alpha$  is the discount factor introduced to resolve the infinite-horizon issue, and  $(0 < \alpha < 1)$  and  $r(t+1)$  are external reinforcement signal values from time  $t+1$ .

The evaluation network and the action network both use the gradient descent rule in their weight renewal.

In evaluation network, when the prediction error at the evaluation network is defined as (4), the target function that requires minimization becomes (5).

$$e_c(t) = \alpha J(t) - [J(t-1) - r(t)] \tag{4}$$

$$E_c(t) = \frac{1}{2} e_c^2(t) \tag{5}$$

In action network, learning takes place through the difference between  $U_c$ , which is the action network's control target, and  $J(t)$ , which has been approximated at the evaluation network.

$$e_a(t) = J(t) - U_c \tag{6}$$

When the difference between  $J(t)$  and  $U_c$  are formed as above, the target function that requires minimization becomes the following:

$$E_a(t) = \frac{1}{2} e_a^2(t) \tag{7}$$

### 3 Simulation

#### 3.1 Simulation Setup

NDP and SOM-applied controller performance in a two-stage inverted pendulum's postural control was tested through a simulation. Fig.2 is a simplified diagram of the two-stage pendulum system used in the simulation. Similar to common two-stage inverted pendulum systems, the system used in the simulation has two poles connected on top of a cart that travels along a limited track. The poles maintain the vertical, positioning the cart to the middle of the track. Table 1 show the parameters used in the simulation's two-stage inverted pendulum system and their values.

From a two-stage inverted pendulum system, six states can be observed; which are: the cart's position  $x_c(t)$ , the angle of the poles from horizontal level  $\theta_1(t), \theta_2(t)$ , the cart's speed  $\dot{x}_c(t)$ , and the poles' angular speed  $\dot{\theta}_1(t), \dot{\theta}_2(t)$ . Range of the cart track is  $[-2.4m +2.4m]$ , and the two poles' range of control is all together  $[-12^\circ +12^\circ]$ .

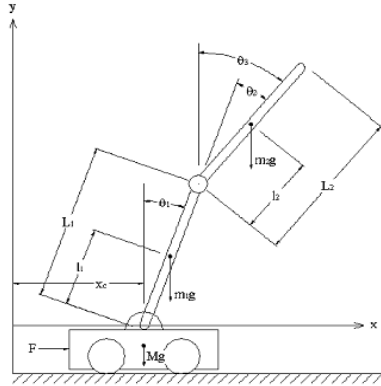


Fig. 2. Two-stage Inverted Pendulum on a cart system

Control is considered as a success if the two poles are maintained inside the range of control for two minutes and the cart do not touch the track's edges. Failure signal used as the reinforcement signal is as shown in equation (8).

$$\text{failure signal} = \begin{cases} -1, & \text{if } |\theta_1| > 12^\circ \text{ or } |\theta_2| > 12^\circ \text{ or } |x_c| > 2.4m \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

**Table 1.** Parameters of Model Equation

parameter	Definition	value
$M$	Mass of cart	0.600 kg
$m_1$	Mass of lower pole	0.3788 kg
$M_2$	Mass of upper pole	0.0722 kg
$L_1$	Length of lower pole	0.42 m
$L_2$	Length of upper pole	0.38 m
$l_1$	Length to center of mass of lower pole	0.299 m
$l_2$	Length to Center of mass of upper pole	0.189 m
$x_c$	Position of cart	[-2.4 2.4] m
$\theta_1$	Angle of lower pole	[-12 12] °
$\theta_2$	Angle of upper pole	[-12 12] °
$g$	gravity acceleration	9.8 m/s <sup>2</sup>
$F$	Force applied to cart	±10 N

The six states have to be used as SOM inputs. But cart speed and angular speed of the poles are sensitive to errors generated from classification using the SOM. Therefore, only the cart's position and the poles' horizontal angle were classified using the SOM. In consideration for the controllable range, the SOM was formed as a three-dimension (10x15x15). Observed inverted pendulum states were then classified using the SOM and used as inputs of the action and evaluation networks.

### 3.2 Simulation Results

100 two-stage inverted pendulum system simulations were run for each of the following cases: using only the TD(0) learning method, using only the NDP learning method, and using the NDP learning method in conjunction with the SOM. Table 2 lists the results from the simulations.

**Table 2.** Simulation results

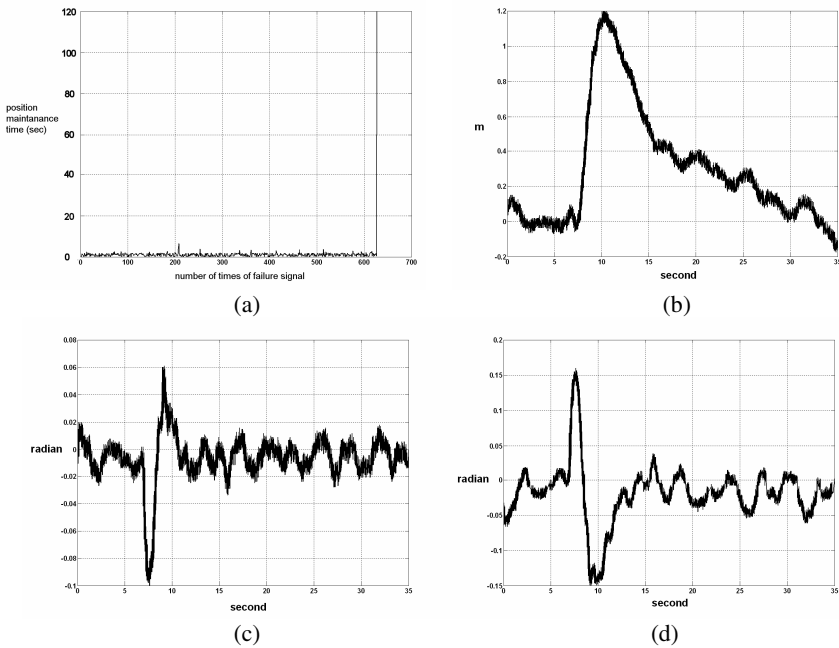
Case description	average number of occurrences of failure signal until success of learning	Success rate of learning	Average time until success of learning
TD(0) algorithm	Over 10000 times	0%	
NDP algorithm	599.22 times	95%	724.5 sec.
NDP algorithm with SOM	508.72 times	95%	643.85 sec.

The case of using TD(0) was limited to 10,000 failure occurrences. According to the simulation results, this case failed to produce a single learning success within its 10,000 failure signal limit in all 100 simulations. After increasing the failure limit to 32,767, the case did produce learning success after over 20,000 failure signals. Conversely, the case of using the NDP learning method only averaged a learning success after 599.22 failure signals. Average time it took for the learning success



724.5 seconds. Such results reaffirm NDP learning method's superior efficiency over the TD(0) learning method and confirm that selection of an efficient learning algorithm in reinforcement learning controls has a significant impact on performance enhancement. In addition, the findings also confirm that SOM efficiently classifies the object data to be learned by the neural network, while preserving the data's characteristics and thus positively influencing performance enhancement.

Figure 3(a) is the position maintenance time until each failure signal generation in the joint NDP learning method and SOM application case (learning success after 627 failure signals). Figure3(b), Fig.3(c) and Fig.3(d) depict changes in cart position, lower pole angle, and upper pole angle during 35 seconds following the learning success.



**Fig. 3.** (a) Position maintenance time until each failure signal (b) cart position during 35 sec. after successful learning (c) upper pole angle during 35 sec. after successful learning (d) . lower pole angle during 35 sec. after successful learning.

### 4 Conclusion

Previous reinforcement learning researches focused, for the most part, on learning success only. In order to apply the reinforcement learning controller to actual systems, however, a steep learning curve is a requisite. Ergo, this study aimed at enhancing the learning speed in digital computer-utilized reinforcement learning controls, for which neural dynamic programming, a type of heuristic dynamic programming, was opted for as the learning method and self-organizing map was used as a means of data

classification. NDP-utilized learning method was chosen after considering that its use of a method that connects an appropriate action for the given moment's states will lead to accelerated learning speed, and this was verified through simulation test results. Another intention was to accelerate the learning speed by decreasing the volume of the neural network controller's learning data used in reinforcement learning. SOM rearranges units on its grid through competitive learning based on input data, which allows it to preserve the data's distribution characteristics while decreasing the data count. Using SOM the number of states used as the controller's learning data was reduced. Simulation results verified that application of the SOM that was selected under the above-mentioned manner did indeed accelerate the learning speed. In addition, the number of learning attempts until learning success was found to be lower when the SOM was used, indicating stable learning performance and a controller characteristic that can be seen as favorable.

This study presents a method of accelerating reinforcement learning controller's learning and verifies the presented method's effectiveness through simulation tests.

**Acknowledgments.** This work was supported by Kangwon National University BK21 project and the Kangwon Institute of Telecommunications and Information.

## References

1. Richard S. Sutton, and Andrew G. Barto, "Reinforcement Learning : An Introduction," MIT Press, Cambridge, MA, 1998.
2. Charles W. Anderson, "Strategy Learning with Multilayer Connectionist Representations," *Proceedings of the 4th International Workshop on Machine Learning*, pp. 103-114, 1987.
3. Charles W. Anderson, "Learning to Control an Inverted Pendulum Using Neural Network," *IEEE Control Systems Magazine*, Vol. 9, No. 3, pp. 31-37, 1989.
4. Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson, "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, No. 5, 1983.
5. Albus, J. S., "A New Approach to Manipulator control : The Cerebellar Model Articulation Controller(CMAC)," *Journal of Dynamics Systems, Measurement, and Control*, pp. 220-227, 1975.
6. Dean F. Hougen, Maria Gini, and James Slagle, "Partitioning input space for reinforcement learning for control," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1917-1922, April, 1996.
7. Andrew James Smith, "Applicatoin of the self-organising map to reinforcement learning," *In Neural Networks (Special Issue)*, 15, pp. 1107-1124, 2002.
8. Kohonen, T., "Self organising maps," Berlin: Springer
9. Richard S. Sutton, "Learning to predict by the methods of temporal difference," *Machine Learning*, Vol. 3, pp. 9-44, 1988.

# Neural Network Mapping of Magnet Based Position Sensing System for Autonomous Robotic Vehicle

Dae–Yeong Im<sup>1</sup>, Young-Jae Ryoo<sup>1</sup>, Jang-Hyun Park<sup>1</sup>, Hyong-Yeol Yang<sup>2</sup>,  
and Ju-Sang Lee<sup>3</sup>

<sup>1</sup> Department of Control System Engineering, Mokpo National University,  
61 Dorim-ri, Muan-goon, Jeonnam 534-729, Korea  
yjryoo@mokpo.ac.kr

<sup>2</sup> Department of Electrical Engineering&RRC-HECS,  
Chonnam National University, Korea

<sup>3</sup> VC R&D Group, Samsung Gwangju Electronics Co. Ltd., Korea

**Abstract.** In this paper a neural network mapping of magnet based position sensing system for an autonomous robotic vehicle. The position sensing system using magnetic markers embedded under the surface of roadway pavement. An important role of magnetic position sensing is identification of vehicle's location. The magnetic sensor measures lateral distance when the vehicle passes over the magnetic marker. California PATH has developed a table-look-up as an inverse map. But it's requires too many memories to store the vast magnetic field data. Thus we propose the magnetic guidance system with simple mapping using neural network.

## 1 Introduction

In a guidance system, position sensing is an important task for the identification of vehicle's locations, such as the lateral position relative to a lane or a desired trajectory. Technologies developed for identifying the vehicle's location include electrically powered wire, computer vision, magnetic sensing, optical sensing, inertia navigation, and global positioning systems [1-3]. This paper focuses on magnetic sensing system [4-7] that are used for ground vehicle control and guidance.

The magnetic marking scheme has several advantages compared with the electrified wiring scheme. Since it is a passive system, it is simple and does not use any energy. A vision sensing scheme is expensive to acquire and to process the optical images into credible data in real-time in any weather and road conditions. In the positioning reference system, magnets are embedded just under the surface of a road as figure 1 shows. The magnetic marking scheme has several advantages compared with the electrified wiring scheme. Since it is a passive system, it is simple and does not use any energy. A vision sensing scheme is expensive to acquire and to process the optical images into credible data in real-time in any weather and road conditions. In the positioning reference system, magnets are embedded just under the surface of a road as Fig. 1 shows.

The major concern in implementing a magnetic sensing system on roads is the background magnetic field. For magnetic sensing, it is a problem that the magnitude

of the background magnetic field is not small compared to that of the magnet’s magnetic field. The background field may be stable or varying, depending on the specific location or the orientation.

This paper suggests a design of the position sensing system with discusses the related technical issues. In this paper, the position sensing technique is first illustrated in Section 2 by introducing the magnetic patterns produced by a sample magnet. The lateral motion control using a test vehicle is reported in Section 3. The paper concludes with a discussion of these concerns in Section 4.

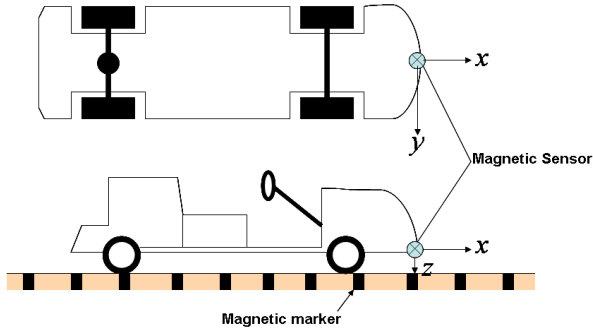


Fig. 1. Position sensing system using magnetic sensor and magnets

## 2 Magnetic Field of Magnet

### 2.1 Analysis of Magnetic Fields of a Magnet

In this section, the comprehensive analysis of the magnetic field of a magnet used for position sensing is presented. Since a typical magnet has the shape of a cylindrical permanent magnet, assume that the magnet is a magnetic dipole. The magnetic field around a magnet can be described using rectangular coordinates as see Figure 2 [8].

$$B_x = \frac{3K_m z x}{(x^2 + y^2 + z^2)^{5/2}} \tag{1}$$

$$B_y = \frac{3K_m y x}{(x^2 + y^2 + z^2)^{5/2}} \tag{2}$$

$$B_z = \frac{K_m (2z^2 - x^2 - y^2)}{(x^2 + y^2 + z^2)^{5/2}} \tag{3}$$

where

$$k_m = \frac{\mu_0 M_T}{4\pi}. \tag{4}$$

( $K_m$ ) is a constant proportional to the strength of the magnet.

$$M_T = \pi b^2 M_0 \tag{5}$$

where ( $M_0$ ) is the magnetization surface charge density, and ( $b$ ) is the radius of the cylindrical permanent magnet.

The polar coordinate equations were used to derive the rectangular coordinate equations. Figure 2 depicts the three-axis components of the magnetic fields using rectangular coordinates. The longitudinal component ( $B_x$ ) is parallel to the line of magnet installation on the road, the vertical component ( $B_z$ ) perpendicular to the surface, and the lateral component ( $B_y$ ) perpendicular to the other two axes. The longitudinal, the lateral, and the vertical component of the magnetic field can be transformed from the polar coordinate equations as:

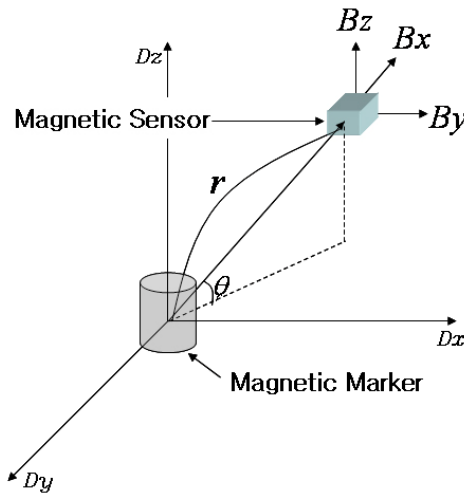
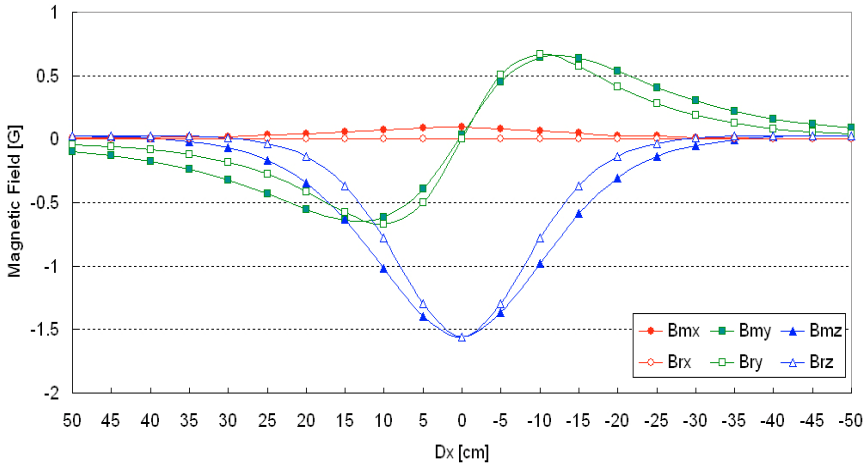


Fig. 2. Magnetic field of magnet

As (1), (2) and (3) show, each component of the field is a function of the strength ( $Km$ ), the longitudinal ( $x$ ), the lateral ( $y$ ), and the vertical ( $z$ ) distance to the sensor.

To verify that these model equations of (1), (2), and (3) represent the physical magnetic fields, the equations are compared with the direct experimental measurement using a ruler on the  $x$ - $y$  test bench table. The magnetic sensor was mounted in a plane above 20cm from the upper end of the magnet.

Figure 3 shows the comparison of the magnetic field components ( $B_{rx}$ ,  $B_{ry}$ ,  $B_{rz}$ ) calculated by the magnetic model equations ( $B_{mx}$ ,  $B_{my}$ ,  $B_{mz}$ ) with those measured by the experiment at various distances to the magnet. The data shows that these model equations are similar with the maximum error of 0.2G along longitudinal ( $B_x$ ), and 0.25G along vertical magnetic field ( $B_z$ ). Thus, the assumption of a dipole magnet is reasonable, and the model equations are useful to represent the magnetic field. However, in practical system, the model equation cannot be used to recalculate the position from the magnetic fields.



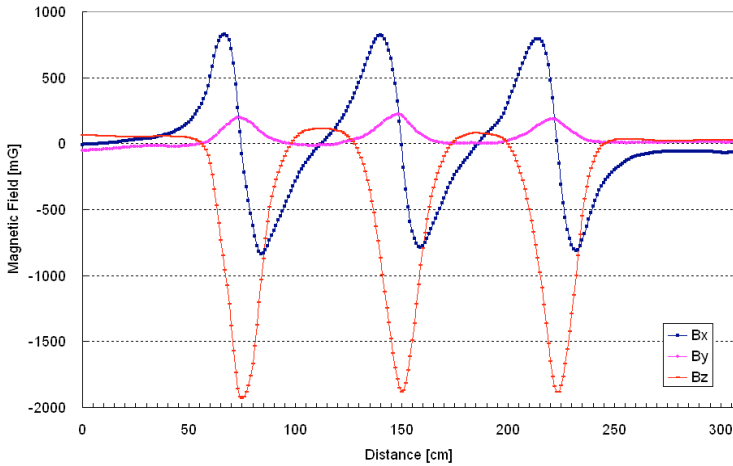
**Fig. 3.** Comparison of the model equations with the physical magnetic field measured using a ruler directly

The experimental data collected from array magnets tests are presented for a comprehensive analysis of the magnetic fields. The first set of data was collected from a static test on a ground. The sample magnet, sold commercially, is made of ceramic material in a cylindrical shape with a diameter of 2.5 cm and a depth of 10 cm. The axis of the cylinder is placed perpendicular to the surface of the magnets. The measurements took place with a magnetic sensor at 20cm heights from the surface of the magnet to acquire a representative map of the magnetic field. For illustration the three-dimensional plots for each component in three orthogonal axes with the sensor at a height 20cm from the upper end of the marker.

Figure 4, shows composite curves with data from the three axes plotted along the distance from the magnet. The curves are generated with the origin at the location of the magnet. Notice how the three components change over space. Each curve represents the measurements at one cross section along the longitudinal axis ( $x$ -axis), while multiple curves in each plot represent the measurement at various locations along the lateral axis ( $y$ -axis). The change of the magnetic fields along these three axes can be clearly identified.

The longitudinal component ( $B_x$ ) rises from zero at the center of the magnet and reaches its peak at a distance about 67cm and 85cm from the magnet, then gradually weakens farther away from the magnet. The peak value for this data set is about 825mG and -835mG. The longitudinal field makes a steep transition near the magnet as it changes its sign. This steep transition becomes meaningful in interpreting the point at which a sensor passes over a magnet location.

The lateral component ( $B_y$ ) reaches its peak at the top of the magnet, and drops down to zero at about 104 mm away from the magnet. The peak value for the data set of lateral component is about 197mG. In any horizontal plane parallel to that of the magnet, as a function of distance from the magnet, the patterns of ( $B_x$ ) and ( $B_y$ ) measurements are similar.



**Fig. 4.** Three-dimensional plots for each component of magnetic field of a magnet with sensor at 20cm high

The vertical field ( $B_x$ ) the strongest right at the top of the magnet, and diminishes to zero at about 75mm away from the magnet. The peak value is above 1934mG. The vertical field quickly drops as it moves away from the magnet. Since the vertical field is the strongest component among the measurements near the magnet, its use will be significant in identifying the closeness of a magnet. The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm.

**2.2 Position Sensing from Magnetic Fields**

Once the magnetic fields are measured, the measured fields at that location will be used to identify the distance to the magnet.

The position of the sensor with respect to the magnets can make a inference from the neural network. The neural network trains the inverse mapping to estimate the position continuously as long as the magnetic field is strong enough to be sensed Figure 5 shows structure of neural network for position sensing. In the implemented system, the transformation of the measured signals to the distance to the magnet is based on the inverse mapping relationship.

Figure 6 shows a sample of inverse mapping of the measured field strengths at a longitudinal field of 0mG.

The data points are produced with the data from the neural network trained the inverse mapping from the magnetic fields to the distance. Measurements from the magnetic field components produce a distance simply. These curves in Figure 6 constitute a basis for position identification. In the implementation, the curves as seen in Figure 6 are calculated in the signal-processing program. However, in a practical system, it should be noted that the magnitude of the background magnetic field is not small compared to that of the magnet magnetic field.

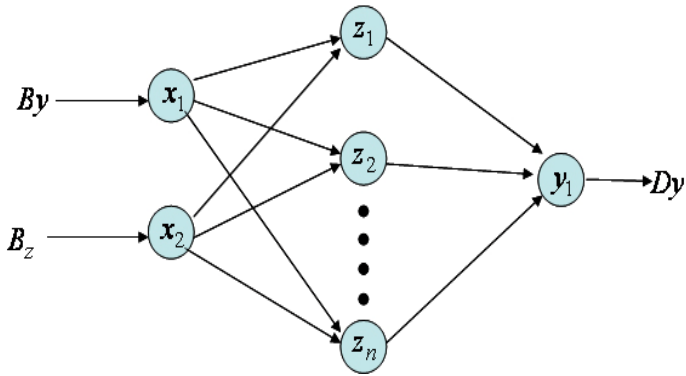


Fig. 5. Structure of neural network for position sensing

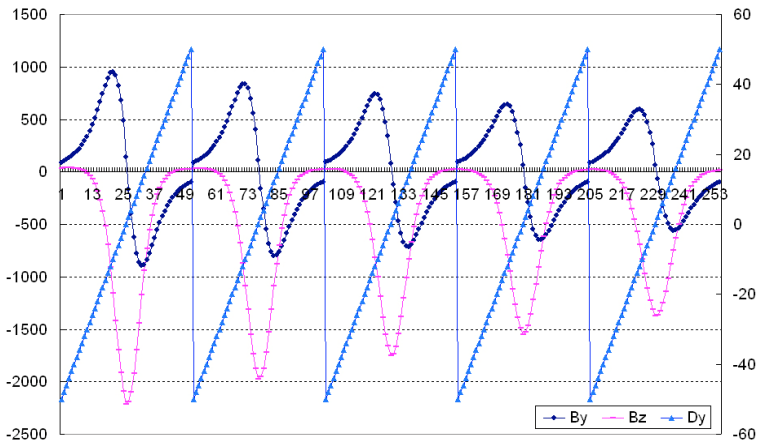


Fig. 6. Position sensing using neural network with sensor at 20 cm high

### 3 Lateral Motion Control

The lateral control tests were performed on a test track. The shape of the test track is as shown in Figure 8. Much of the test track was based upon a previously existing dirt road at the site. The track is approximately 38 meter long. The permanent magnets, with 2.5 cm diameter and 10 cm depth, were installed along the center of the track at one meter spacing.

According to the inverse mapping described in Section 2, the position of the sensor installed in the vehicle relative to the magnet was calculated. The lateral error of the vehicle from the center of the road is produced. A simple and fast PD based control is used to demonstrate the practical steering of the vehicle. The input of the controller is the lateral error, and the output of the controller is the steering angle.





Fig. 7. Photograph of the autonomous robotic vehicle

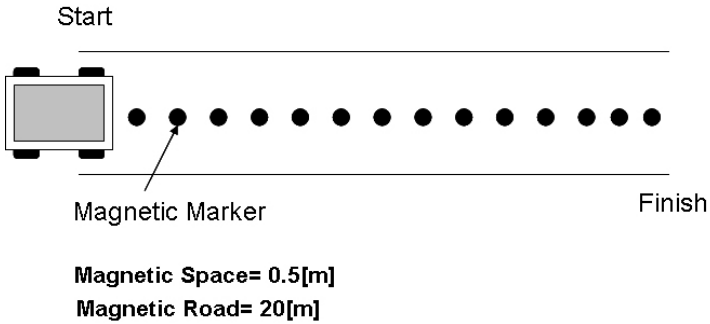


Fig. 8. The layout of the test track

Through the proposed method, the neural network mapping a steering angle is produced. The steering angle remained very smooth. When the steering angle changes from a negative value to a positive value, the vehicle turns from the right and to the left.

## 4 Conclusions

In this paper, a position reference system using a magnetic sensor and magnet was discussed. The field patterns of a sample magnet were first measured to illustrate the basic characteristics of such systems. The properties of the background field were

then examined by conducting experiments at different geometric orientations. The observations from these experiments revealed that indeed there were potential complications that might be caused by the vehicle's position and orientation. These effects must be handled carefully to ensure a robust sensing approach for correct identification of the vehicle's position.

The lateral motion control of the test vehicle shows that it is reliable to sense the vehicle's position from the magnetic field on the real track. The suggested design of position sensing system was working well on a road with high curvature. When the deployment of magnetic sensing systems for position reference increases, these subjects deserve elaborated studies and thorough analysis.

## References

1. Shumeet Baluja, Evolution of an artificial neural network based autonomous land vehicle controller, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 26, no. 3, (1996) 450-463.
2. Kevin M. Passio, Intelligent control for autonomous vehicle, *IEEE Spectrum*, (1995) 55-62.
3. Ronald K. Jurgen, Smart cars and highways go global, *IEEE Spectrum*, (1991) 26-36.
4. James G. Bender, An overview of system studies of automated highway systems, *IEEE Transactions on vehicular technology*, vol. 40, no. 1, (1991).
5. Seibum B. Choi, The design of a look-down feedback adaptive controller for the lateral control of the front-wheel-steering autonomous vehicles, *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, (2000) 2257-2269.
6. Ching-Yao Chan, Magnetic sensing as a position reference system for ground vehicle control, *IEEE Trans. Inst. and Meas.*, vol. 51, no. 1, (2002) 43-52.
7. Han-Shue Tan, J. Guldner, S. Patwardhan, Chieh Chen, and B. Bougler, Development of an automated steering vehicle based on road magnets - a case study of mechatronic system design, *IEEE/ASME Transactions on Mechatronics*, vol. 4, no. 3, (1999) 258-272.
8. E. S. Shire, *Classical electricity and magnetism*, Cambridge, U.K.: Cambridge University Press, (1960).

# Application of Fuzzy Integral Control for Output Regulation of Asymmetric Half-Bridge DC/DC Converter

Gyo-Bum Chung

Department of Electrical Engineering, Hongik University  
Jochiwon, Chungnam, 339-701, Korea  
gbchung@hongik.ac.kr

**Abstract.** This paper considers the problem of regulating the output voltage of an asymmetric half-bridge (AHB) DC/DC converter via fuzzy integral control. First, we model the dynamic characteristics of the AHB DC/DC converter with the state-space averaging method, and after introducing an additional integral state of the output regulation error, we obtain the Takagi-Sugeno (TS) fuzzy model for the augmented system. Second, the concept of the parallel distributed compensation is applied to the design of the TS fuzzy integral controller, in which the state feedback gains are obtained by solving the linear matrix inequalities (LMIs). Finally, numerical simulations are performed for the considered application.

## 1 Introduction

Recently, with rapid progress made in power semi-conductor applications, the needs for high-performance control have dramatically increased in the area. In particular, the methods of LQG control,  $H_\infty$  control and fuzzy control have been successfully applied to achieve the stability, robustness, and the output-regulation for switching power converters such as AHB (asymmetric half-bridge) DC/DC converter [1], boost converter [2,3], and buck converter [4,5]. Among the results, the TS (Takagi-Sugeno) fuzzy integral control approach which is recently proposed by Lian *et al.* [4] turns out to be very promising, since it guarantees the stable output-regulation as well as the robustness and disturbance rejection capability. Motivated by the recent successful applications of the TS fuzzy integral control method to various type of converters [4,5], this paper considers the problem of applying the method to the output-regulation of the AHB DC/DC converter. The AHB DC/DC converter has been known to have many advantages in the area of the power conversion [6], and due to the nonlinear characteristics of the converter system, maintaining its output voltage constant regardless of the output load changes has been known to be a difficult task. The remaining parts of this paper are organized as follows: Sect. 2 presents the modeling process for the AHB DC/DC converter. Section 3 describes how to apply the integral TS fuzzy control to the output regulation of the converter. Finally, in Sect. 4 and 5, simulation results and concluding remarks are given, respectively.

## 2 Modeling of Asymmetric Half-Bridge DC/DC Converter

Fig. 1 shows the AHB DC/DC converter with fuzzy integral controller to regulate the output voltage  $V_o$  for a resistive load  $R$ . Switch  $S_1$  with duty ratio  $d$  and switch  $S_2$  with duty ratio  $(1-d)$  operate complementarily in a constant switching period  $T$  [6].  $N_0$  is the number of turns on the primary winding of transformer  $T_x$ .  $N_1$  and  $N_2$  are the numbers of turns on the secondary windings. For the analysis

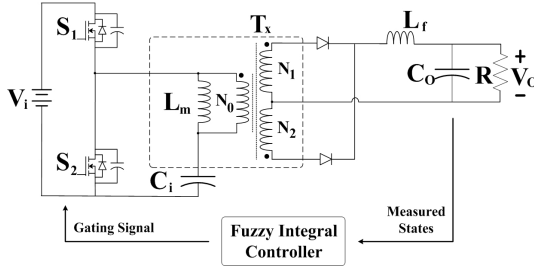


Fig. 1. Power circuit of the AHB DC/DC converter

of the converter operation, the parasitic resistances of inductor and capacitors are explicitly considered, while the dead time between  $S_1$  and  $S_2$ , the leakage inductance of transformer  $T_x$ , and diode voltage drop are neglected. Also, it is assumed that  $n_1 = N_1/N_0 = n_2 = N_2/N_0 = n$ .

When switch  $S_1$  is on and switch  $S_2$  is off for  $1 \leq t \leq dT$ , the converter in Fig.1 can be described by

$$\begin{aligned} \dot{x}_p &= A_1 x_p + B_1 V_i, \\ y &= C_1 x_p, \end{aligned} \tag{1}$$

where

$$\begin{aligned} x_p &\triangleq [v_{C_i} \ i_{L_m} \ i_{L_F} \ v_{C_o}]^T, \\ A_1 &= \begin{bmatrix} 0 & 1/C_i & n_1/C_i & 0 \\ -1/L_m & -R_i/L_m & -R_i n_1/L_m & 0 \\ -n_1/L_F & -n_1 R_i/L_F & a_{33}/L_F & -a_{34}/L_F \\ 0 & 0 & a_{34}/C_o & -a_{44}/C_o \end{bmatrix}, \\ a_{33} &= -(n_1^2 R_i + R_C R)/(R_C + R) + R_F, \\ a_{34} &= R/(R_C + R), \quad a_{44} = 1/(R_C + R), \\ B_1 &= [0 \ 1/L_m \ n_1/L_F \ 0]^T, \\ C_1 &= [0 \ 0 \ R_C a_{34} \ a_{34}]. \end{aligned} \tag{2}$$

When switch  $S_1$  is off and switch  $S_2$  is on for  $(1-d)T \leq t \leq T$ , the converter in Fig. 1 can be described by

$$\begin{aligned} \dot{x}_p &= A_2 x_p + B_2 V_i, \\ y &= C_2 x_p, \end{aligned} \tag{3}$$

where

$$\begin{aligned}
 A_2 &= \begin{bmatrix} 0 & 1/C_i & n_2/C_i & 0 \\ -1/L_m & -R_i/L_m & R_i n_2/L_m & 0 \\ n_2/L_F & n_2 R_i/L_F & a_{2,33}/L_F & -a_{34}/L_F \\ 0 & 0 & a_{34}/C_o & -a_{44}/C_o \end{bmatrix}, \\
 a_{2,33} &= -(n_2^2 R_i + R_C R/(R_C + R) + R_F), \\
 B_2 &= [0 \ 0 \ 0 \ 0]^T, \\
 C_2 &= [0 \ 0 \ R_C a_{34} \ a_{34}].
 \end{aligned} \tag{4}$$

From the above, we can obtain the following state-space-averaged representation of the AHB DC/DC converter:

$$\begin{aligned}
 \dot{x}_p &= A_a x_p + B_a V_i \\
 &= [dA_1 + (1-d)A_2]x_p + [dB_1 + (1-d)B_2]V_i, \\
 y &= C_a x_p \\
 &= [dC_1 + (1-d)C_2]x_p,
 \end{aligned} \tag{5}$$

where

$$\begin{aligned}
 A_a &= \begin{bmatrix} 0 & 1/C_i & (2d-1)n/C_i & 0 \\ -1/L_m & -R_i/L_m & -(2d-1)R_i n/L_m & 0 \\ -(2d-1)n/L_F & -(2d-1)nR_i/L_F & a_{2,33}/L_F & -a_{34}/L_F \\ 0 & 0 & a_{34}/C_o & -a_{44}/C_o \end{bmatrix}, \\
 B_a &= [0 \ d/L_m \ nd/L_F \ 0]^T, \\
 C_a &= [0 \ 0 \ R_C a_{34} \ a_{34}].
 \end{aligned} \tag{6}$$

### 3 Design of Integral TS Fuzzy Control Via LMIs

In this section, we are concerned with an integral TS fuzzy control approach to regulating the AHB DC/DC converter. Here, we follow the strategy of Lian *et al.* [4] with a slight modification to design the fuzzy controller, and the design strategy is composed of the following steps: First, the additional integral error signal is introduced to form the augmented system consisting of the converter dynamics and error dynamics. Then, the standard TS fuzzy model is established with the new coordinates centered at the regulated points. Finally, the concept of parallel distributed compensation [7] is applied to design a TS fuzzy controller, in which the state feedback gains are obtained by solving LMIs. In the following, we will explain the design strategy in a step-by-step manner:

According to the modeling of Sect. 2, one can express the considered converter system in the following general form:

$$\begin{aligned}
 \dot{x}_p(t) &= f(x_p(t), d(t)), \\
 y(t) &= h(x_p(t)),
 \end{aligned} \tag{7}$$

where  $x_p(t) \in R^4$ ,  $d(t) \in R$ ,  $y(t) \in R$  are the state, the control input, and the output, respectively. For this systems, let  $r \in R$  be a constant desirable reference,

and we want to design a controller achieving the goal  $y(t) \rightarrow r$  as  $t \rightarrow \infty$ . For this goal, we will use the integral TS fuzzy control [4], which belongs to the integral-type controllers, thus can not only achieve zero steady-state regulation error but also be robust to uncertainty and disturbance. In the integral control, a state variable  $x_e(t)$  is additionally introduced to account for the integral of the output regulation error, and thus it satisfies

$$\dot{x}_e(t) = r - y(t) . \tag{8}$$

Incorporating the dynamics for  $x_e$  into the original nonlinear system (7), one can obtain the following augmented state equation:

$$\begin{aligned} \dot{x}_p(t) &= f(x_p(t), d(t)) , \\ \dot{x}_e(t) &= r - h(x_p(t)) . \end{aligned} \tag{9}$$

Here, the output regulation can be achieved by stabilizing the whole system around an equilibrium state which can yield  $y = h(x_p)$  being equal to  $r$ . For this, let  $\bar{x}_p \in R^4$  and  $\bar{d} \in R$  be such that

$$\begin{aligned} f(\bar{x}_p, \bar{d}) &= 0 , \\ r - h(\bar{x}_p) &= 0 , \end{aligned} \tag{10}$$

and let  $\tilde{x}_p(t)$  and  $\tilde{d}(t)$  be the new coordinates centered at the regulated points, *i.e.*,  $\tilde{x}_p(t) \triangleq x_p(t) - \bar{x}_p$  and  $\tilde{d}(t) \triangleq d(t) - \bar{d}$ . Since  $\bar{x}_e$  cannot be determined from (10), one may treat  $\bar{x}_e$  as a design parameter that can be freely chosen. In this paper,  $\bar{x}_e = 0$  is chosen for simplicity, and  $\tilde{x}_e(t)$  is defined accordingly. Note that equation (9) can now be expressed using the new coordinates as follows:

$$\begin{aligned} \dot{\tilde{x}}_p(t) &= f(\bar{x}_p + \tilde{x}_p(t)) , \\ \bar{d} + \tilde{d}(t) &= f_0(\tilde{x}_p(t), \tilde{d}(t)) , \\ \dot{\tilde{x}}_e(t) &= r - h(\bar{x}_p + \tilde{x}_p(t)) = h_0(\tilde{x}_p(t)) . \end{aligned} \tag{11}$$

Also note that in (11), the newly defined functions,  $f_0$  and  $h_0$ , satisfy  $f_0(0, 0) = 0$  and  $h_0(0) = 0$ , respectively. Now from the modeling results of the previous section, it is observed that the augmented system (11) for the AHB DC/DC converter can be represented by the TS fuzzy model, in which the  $i$ -th rule has the following form:

*Plant Rule i:*

$$\text{IF } z_1(t) \text{ is } F_1^i \text{ and } \dots z_g(t) \text{ is } F_g^i, \text{ THEN } \dot{\tilde{x}}(t) = A_i \tilde{x}(t) + B_i \tilde{d}(t) , \tag{12}$$

where  $i = 1, \dots, m$ . Here,  $\tilde{x}(t) \triangleq [\tilde{x}_p^T(t) \ \tilde{x}_e(t)]^T$  is the state vector;  $z_j(t), j = 1, \dots, g$  are premise variables each of which is selected from the entries of  $x_p(t)$ ;

$F_j^i, j = 1, \dots, g, i = 1, \dots, m$  are fuzzy sets;  $m$  is the number of IF-THEN rules, and  $(A_i, B_i)$  is the  $i$ -th local model of the fuzzy system. Utilizing the usual inference method, one can obtain the following state equation for the TS fuzzy system [7]:

$$\dot{\hat{x}}(t) = \sum_{i=1}^m \mu_i(x_p(t)) \{A_i \hat{x}(t) + B_i \tilde{d}(t)\} , \tag{13}$$

where the normalized weight functions  $\mu_i(x_p(t)) \triangleq w_i(x_p(t)) / \sum_i w_i(x_p(t))$  with  $w_i(x_p(t)) \triangleq \prod_{j=1}^g F_j^i(x_p(t))$  satisfy

$$\mu_i(x_p(t)) \geq 0, \quad i = 1, \dots, m , \tag{14}$$

and

$$\sum_{i=1}^m \mu_i(x_p(t)) = 1 \text{ for any } t \geq 0 . \tag{15}$$

For simplicity, we will denote the normalized weight function  $\mu_i(x_p(t))$  by  $\mu_i$  from now on. According to the concept of the parallel distributed compensation [7], the TS fuzzy system (13) can be effectively controlled by the controller characterized by the following fuzzy IF-THEN rules:

*Controller Rule i:*

$$\text{IF } z_1(t) \text{ is } F_1^i \text{ and } \dots z_g(t) \text{ is } F_g^i, \text{ THEN } \tilde{d}(t) = -K_i \hat{x}(t) , \tag{16}$$

where  $i = 1, \dots, m$ . Note that the IF part of the above controller rule shares the same fuzzy sets with that of (12). The usual inference method for the TS fuzzy model yields the following TS fuzzy controller [7]

$$\tilde{d}(t) = - \sum_{i=1}^m \mu_i K_i \hat{x}(t) , \tag{17}$$

and plugging (17) into (13) yields the closed-loop system represented as

$$\dot{\hat{x}}(t) = \sum_{i=1}^m \sum_{j=1}^m \mu_i \mu_j (A_i - B_i K_j) \hat{x}(t) . \tag{18}$$

Here, the state feedback gains  $K_i$  can be found by solving the LMIs of the following theorem, which is obtained by combining Theorem 1 of [4] together with Theorem 2.2 of [8]:

*Theorem:* Let  $D$  be a diagonal positive-definite matrix. The closed-loop (18) can be exponentially stabilized via the controller (17) with  $K_j = M_j X^{-1}$  if there exists  $X = X^T > 0$  and  $M_1, \dots, M_m$  satisfying the following LMIs:

$$\begin{aligned} N_{ii}(Y) &< 0, \quad i = 1, \dots, m , \\ \frac{1}{m-1} N_{ii}(Y) + \frac{1}{2} (N_{ij}(Y) + N_{ji}(Y)) &< 0 , \quad 1 \leq i \neq j \leq m , \end{aligned} \tag{19}$$

where

$$\begin{aligned}
 Y &\triangleq (X, M_1, \dots, M_m), \\
 N_{ij}(Y) &\triangleq \begin{bmatrix} A_i X + X A_i^T - B_i M_j - M_j^T B_i^T & X D^T \\ DX & -X \end{bmatrix}.
 \end{aligned}
 \tag{20}$$

### 4 A Design Example and Simulation

In this section, we present an example of the TS fuzzy control approach described above applied to the problem of regulating the output voltage of the AHB DC/DC converter. We consider the converter with the system parameters of Table 1 [6], and its equilibrium points satisfying (10) are as follows:  $\bar{x}_p = [\bar{v}_{Ci} \ \bar{i}_{Lm} \ \bar{i}_{LF} \ \bar{v}_{Co}]^T = [57 \ 0.4667 \ 5.018 \ 13.0469]^T$  and  $\bar{d} = 0.19$ .

**Table 1.** Parameters of the the AHB DC/DC converter [6]

Parameters	Value	Unit
Input Voltage, $V_i$	300	V
Input Capacitor, $C_i$	0.82	$\mu F$
Parasitic Resistance for $C_i, R_i$	0.74	$\Omega$
Magnetizing Inductance, $L_m$	198	$\mu H$
Output Inductance, $L_F$	18	$\mu H$
Parasitic Resistance for $L_F, R_F$	0.15	$\Omega$
Output Capacitance, $C_o$	880	$\mu F$
Parasitic Resistance for $C_o, R_C$	0.0025	$\Omega$
Output Resistance, $R$	2.6	$\Omega$
Transformer Turn Ratio, $n$	0.15	
Switching Frequency, $f_s$	100	$kHz$
Nominal Output Voltage, $V_o$	13	V

As mentioned before, substituting  $x_p(t) = \bar{x}_p + \tilde{x}_p(t)$ ,  $x_e(t) = \bar{x}_e + \tilde{x}_e(t)$ , and  $d(t) = \bar{d} + \tilde{d}(t)$  into equations (5) and (8) yields the augmented system, which can be represented as the TS fuzzy IF-THEN rules (12). Here, with  $l_1 = 57$ ,  $l_2 = 0.4$ , and  $l_3 = 5.018$ , the membership functions of (12) are defined as follows:

$$\begin{aligned}
 F_1^1(v_{Ci}) &= F_1^2(v_{Ci}) = F_1^3(v_{Ci}) = F_1^4(v_{Ci}) = 1/2 + \tilde{v}_{Ci}/(2l_1), \\
 F_1^5(v_{Ci}) &= F_1^6(v_{Ci}) = F_1^7(v_{Ci}) = F_1^8(v_{Ci}) = 1/2 - \tilde{v}_{Ci}/(2l_1), \\
 F_2^1(i_{Lm}) &= F_2^2(i_{Lm}) = F_2^5(i_{Lm}) = F_2^6(i_{Lm}) = 1/2 + \tilde{i}_{Lm}/(2l_2), \\
 F_2^3(i_{Lm}) &= F_2^4(i_{Lm}) = F_2^7(i_{Lm}) = F_2^8(i_{Lm}) = 1/2 - \tilde{i}_{Lm}/(2l_2), \\
 F_3^1(i_{LF}) &= F_3^3(i_{LF}) = F_3^5(i_{LF}) = F_3^7(i_{LF}) = 1/2 + \tilde{i}_{LF}/(2l_3), \\
 F_3^2(i_{LF}) &= F_3^4(i_{LF}) = F_3^6(i_{LF}) = F_3^8(i_{LF}) = 1/2 - \tilde{i}_{LF}/(2l_3).
 \end{aligned}
 \tag{21}$$

Also, the resultant local models  $(A_i, B_i)$ ,  $i = 1, \dots, 8$  obtained by plugging the parameter values of Table 1 are as follows:



$$\begin{aligned}
 A_1 = \dots = A_8 = 10^6 \cdot & \begin{bmatrix} 0 & 1.2195 & -0.1134 & 0 & 0 \\ -0.0051 & -0.0037 & 0.0003 & 0 & 0 \\ 0.0052 & 0.0038 & -0.0094 & -0.0555 & 0 \\ 0 & 0 & 0.0011 & -0.0004 & 0 \\ 0 & 0 & -0.0250 & -0.9990 & 0 \end{bmatrix}, \\
 B_1 = 10^6 \cdot & \begin{bmatrix} 3.665 \\ 1.504 \\ 1.374 \\ 0 \\ 0 \end{bmatrix}, \quad B_2 = 10^6 \cdot \begin{bmatrix} 3.665 \\ 1.504 \\ 1.707 \\ 0 \\ 0 \end{bmatrix}, \quad B_3 = 10^6 \cdot \begin{bmatrix} 3.665 \\ 1.504 \\ 1.381 \\ 0 \\ 0 \end{bmatrix}, \\
 B_4 = 10^6 \cdot & \begin{bmatrix} 3.665 \\ 1.504 \\ 1.715 \\ 0 \\ 0 \end{bmatrix}, \quad B_5 = 10^6 \cdot \begin{bmatrix} 0.007 \\ 1.515 \\ 1.374 \\ 0 \\ 0 \end{bmatrix}, \quad B_6 = 10^6 \cdot \begin{bmatrix} 0.007 \\ 1.515 \\ 1.707 \\ 0 \\ 0 \end{bmatrix}, \\
 B_7 = 10^6 \cdot & \begin{bmatrix} 0.007 \\ 1.515 \\ 1.381 \\ 0 \\ 0 \end{bmatrix}, \quad B_8 = 10^6 \cdot \begin{bmatrix} 0.007 \\ 1.515 \\ 1.715 \\ 0 \\ 0 \end{bmatrix}.
 \end{aligned} \tag{22}$$

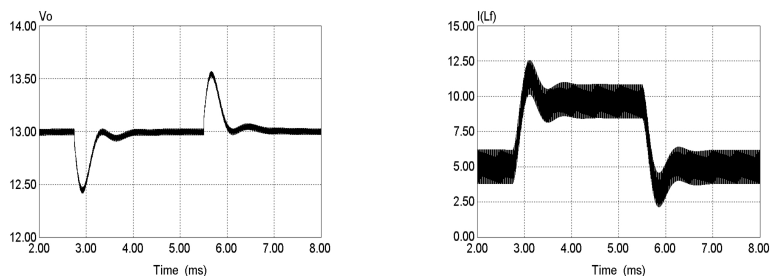
By solving the LMIs (19) via the LMI control toolbox [9] of Matlab with  $D = \text{diag}\{10, 10, 10, 10, 50\}$ , we obtained the following state feedback gains.

$$\begin{aligned}
 K_1 &= [0.1329 \cdot 10^{-4} \quad 0.0006 \quad 0.0009 \quad 0.0100 \quad -61.9090], \\
 K_2 &= [0.1215 \cdot 10^{-4} \quad 0.0005 \quad 0.0009 \quad 0.0099 \quad -60.9990], \\
 K_3 &= [0.1327 \cdot 10^{-4} \quad 0.0006 \quad 0.0009 \quad 0.0100 \quad -61.8916], \\
 K_4 &= [0.1213 \cdot 10^{-4} \quad 0.0005 \quad 0.0009 \quad 0.0099 \quad -60.9759], \\
 K_5 &= [0.1115 \cdot 10^{-4} \quad 0.0029 \quad 0.0017 \quad 0.0193 \quad -116.3605], \\
 K_6 &= [0.0710 \cdot 10^{-4} \quad 0.0027 \quad 0.0017 \quad 0.0187 \quad -112.3930], \\
 K_7 &= [0.1106 \cdot 10^{-4} \quad 0.0029 \quad 0.0017 \quad 0.0193 \quad -116.2791], \\
 K_8 &= [0.0700 \cdot 10^{-4} \quad 0.0027 \quad 0.0017 \quad 0.0187 \quad -112.2954].
 \end{aligned} \tag{23}$$

Numerical simulations were carried out with PSIM [10] for the designed TS integral fuzzy controller, and their results were shown in Fig. 2. Here, the load resistance changed from 2.6[Ω] to 1.3[Ω] at 2.75[ms] and back to 2.6[Ω] at 5.5[ms]. From the figure, one can see that the TS fuzzy integral controller regulates the output voltage  $V_o$  at 13[V] smoothly in 1[ms] as the inductor current  $i_{L_F}$  changes to the values required by the load.

### 5 Concluding Remarks

In this paper, the TS fuzzy integral control approach proposed by Lian *et al.* [4] was applied to the regulation of the output voltage of the AHB DC/DC



**Fig. 2.** Simulation results for the TS fuzzy integral control approach applied to the AHB DC/DC converter: Output voltage  $V_o$  (left) and inductor current  $i_{L_F}$  (right)

converter. After modeling the dynamic characteristics of the AHB DC/DC converter with state-space averaging method and additionally introducing the integral state of the output regulation error, we obtained the TS fuzzy model for the augmented system in the new coordinate centered in equilibrium points. Then, the concept of the parallel distributed compensation was employed, and the state feedback gains of the TS fuzzy integral controller were obtained by solving the linear matrix inequalities (LMIs). Since LMIs can be solved efficiently within a given tolerance by the interior point methods, the LMI-based design is quite effective in practice. Simulations in the time-domain utilizing PSIM program showed that the performance of the designed TS fuzzy integral controller is satisfactory. Further investigations yet to be done include the extension of the considered method toward the use of piecewise quadratic Lyapunov functions.

## References

1. Park, J.H., Zolghadri, M.R., Kimiaghali, B., Homaifar, A., Lee, F.C.: LQG Controller for asymmetrical half-bridge converter with range winding. In: Proceedings of 2004 IEEE International Symposium on Industrial Electronics (2004) 1261–1265
2. Lam, H.K., Lee, T.H., Leung, F.H.F., Tam, P.K.S.: Fuzzy control of DC-DC switching converters: stability and robustness analysis. In: Proceedings of 27th Annual Conf. of the IEEE Industrial Electronics Society (2001) 899–902
3. Naim, R., Weiss G., Ben-Yaakov, S.:  $H^\infty$  control applied to boost power converters. IEEE Transactions on Power Electronics **12** (1997) 677–683
4. Lian, K.-Y., Liou, J.-J., Huang, C.-Y.: LMI-based integral fuzzy control of DC-DC converters. IEEE Transactions on Fuzzy Systems **14**(1) (2006) 71–80
5. Chiang, T.-S., Chiu, C.-S., Liu, P.: Robust fuzzy integral regulator design for a class of affine nonlinear systems. IEICE Transactions on Fundamentals **E89-A**(4) (2006) 1100–1107
6. Korotkov, S., Meleshin, V., Nemchinov, A., Fraidlin, S.: Small-signal modeling of soft-switched asymmetrical half-bridge DC/DC converter. In: Proceedings of IEEE APEC95 (1995) 707–711
7. Tanaka, K., Wang, H.O.: Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequalities Approach. John Wiley & Sons, New York (2001)

8. Tuan, H.D., Apkarian, P., Narikiyo, T., Yamamoto, Y.: Parameterized linear matrix inequality techniques in fuzzy control system design. *IEEE Transactions on Fuzzy Systems* **9(2)** (2001) 324–332
9. Gahinet, P., Nemirovski, A., Laub, A.J., Chilali, M.: *LMI Control Toolbox*, MathWorks Inc., Natick, MA (1995)
10. PSIM User Manual. <http://www.powersimtech.com/manual/psim-manual.pdf> Powersim Inc. (2006)

# Obtaining an Optimum PID Controller Via Adaptive Tabu Search

Deacha Puangdownreong<sup>1</sup> and Sarawut Sujitjorn<sup>2</sup>

<sup>1</sup>Department of Electrical Engineering, Faculty of Engineering,  
South-East Asia University, Bangkok, 10160, Thailand  
dp@sau.ac.th

<sup>2</sup>School of Electrical Engineering, Suranaree University of Technology,  
Nakhon Ratchasima, 30000, Thailand

**Abstract.** An application of the Adaptive Tabu Search (ATS), an intelligent search method in industrial control domain, is presented. The ATS is used to search for the optimum controller's parameters denoted as proportional, integral, and derivative gains. The obtained controllers are tested against some hard-to-be-controlled plants. The results obtained are very satisfactory.

## 1 Introduction

The use of proportional-integral-derivative (PID) controllers for industrial applications was first introduced in 1939 [1]. Due to ease of use and simple realization, PID controllers have been increasingly employed in the feedback control system over decades. To obtain appropriate controller's parameters, one can proceed with available analytical design methods or tuning rules. Mostly the analytical design methods assume known plant models [2],[3],[4], while the tuning rules assume known process responses [5],[6], and known plant models [7]. Those analytical design methods and tuning rules, however, have some particular conditions concerning the plant models, such as dead time or transport lag, fast and slow poles, real and complex conjugated zeros and poles, as well as unstable poles, etc. These conditions make the design methods and tuning rules non-general. In 2000, Åström, et al. [8] proposed a collection of systems which are suitable for testing PID controllers. These collected systems are considered benchmark problems or batch of test examples for evaluating some proposed PID controller design methods.

To date, artificial intelligent (AI) techniques have been accepted and used for the controller design in industrial control applications. For example, designing of an adaptive PID controller by Genetic Algorithm (GA) [9], self-tuning PID controller by GA [10], and finite-precision PID controller by GA [11]. Although the GA is efficient to find the global minimum of the search space, it consumes too much search time. The adaptive tabu search (ATS) method is an alternative, which has global convergence property [12]. It has also been applied to linear and nonlinear identifications for some complex systems [13]. Thus, the ATS method is expected to be an alternative potential algorithm to obtain an optimum PID controller. In this paper, the ATS method is briefly reviewed and then applied to design PID controllers for benchmark

systems proposed by Åström, et al. [8]. This paper consists of five sections. The problem formulation of PID design is described in Section 2. Section 3 provides a brief of the ATS algorithm. The ATS-based design of the PID controllers is illustrated in Section 4, while Section 5 gives the conclusions.

## 2 Problem Formulation

A conventional control loop is represented by the block diagram in Fig. 1. The PID controller receives the error signal,  $E(s)$ , and generates the control signal,  $U(s)$ , to regulate the output response,  $C(s)$ , referred to the input,  $R(s)$ , where  $D(s)$  is disturbance signal,  $G_p(s)$  and  $G_c(s)$  are the plant and the controller transfer functions, respectively.

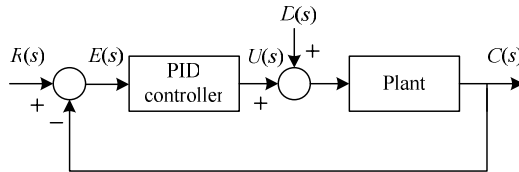


Fig. 1. Conventional control loop

The transfer function of the PID controller is stated in (1), where  $K_p$  is the proportional gain,  $K_i$  is the integral gain, and  $K_d$  is the derivative gain. So, the design problem is simplified to determine the parameters  $K_p$ ,  $K_i$ , and  $K_d$ . The closed loop transfer function with PID controller is given in (2)

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \tag{1}$$

$$\frac{C(s)}{R(s)} = \frac{\left( K_p + \frac{K_i}{s} + K_d s \right) G_p(s)}{1 + \left( K_p + \frac{K_i}{s} + K_d s \right) G_p(s)} \tag{2}$$

The use of AI search techniques to design the PID controller can be represented by the block diagram in Fig. 2. The cost function,  $J$ , the sum of absolute errors between  $R(s)$  and  $C(s)$  as stated in (3), is fed back to the AI tuning block.  $J$  is minimized to find the optimum PID controller’s parameters. In this work, the AI tuning block contains the ATS algorithm

$$J = \sum_{t=0}^T |r(t) - c(t)| \tag{3}$$

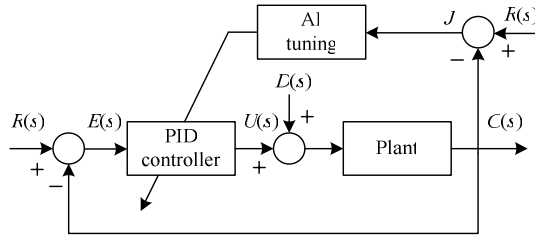


Fig. 2. AI-based PID controller design

### 3 ATS Algorithm

The ATS method [12],[13] is one of the most efficient AI search techniques. It is based on iterative neighborhood search approach for solving combinatorial and nonlinear problems. The Tabu list, one important feature of the method that has first-in-last-out property, is used to record a history of solution movement for leading a new direction that can escape a local minimum trap. In addition, the ATS method has two additional mechanisms, namely back-tracking and adaptive search radius, to enhance its convergence. The ATS algorithm is summarized, step-by-step, as follows.

- Step 1) Initialize a search space,  $count$  and  $count_{max}$  (maximum search round).
- Step 2) Randomly select an initial solution  $x_0$  from the search space. Let  $x_0$  be a current local minimum.
- Step 3) Randomly generate  $N$  solutions around  $x_0$  within a certain radius  $R$ . Store the  $N$  solutions, called neighborhood, in a set  $X$ .
- Step 4) Evaluate a cost function of each member in  $X$ . Set  $x_1$  as a member that gives the minimum cost in  $X$ .
- Step 5) If  $x_1 < x_0$ , put  $x_0$  into the Tabu list and set  $x_0 = x_1$ , otherwise, store  $x_1$  in the Tabu list instead.
- Step 6) Activate the back-tracking mechanism, when solution cycling occurs.
- Step 7) If the termination criteria:  $count \geq count_{max}$ , or desired specifications are met, then stop the search process.  $x_0$  is the best solution, otherwise go to Step 8.
- Step 8) Activate the adaptive search radius mechanism, when a current solution  $x_0$  is relatively close to a local minimum to refine searching accuracy.
- Step 9) Update  $count$ , and go to Step 2.

The back-tracking mechanism described in Step 6 is active when the number of solution cycling is equal to the maximum solution-cycling allowance. This mechanism selects an already visited solution stored in the Tabu list as an initial solution for the next search round to enable a new search path that could escape the local deadlock towards a new local minimum. For the adaptive search radius mechanism described in Step 8, it is invoked when a current solution is relatively close to a local minimum. The radius is thus decreased in accordance with the best cost function found so far. The less the cost function, the smaller the radius. With these two features, a sequence of solutions obtained by the ATS method rapidly converges to the global minimum. The following recommendations are useful for setting the initial values of search parameters:

- i) the initial search radius,  $R$ , should be 7.5-15.0% of the search space radius,
- ii) the number of neighborhood members,  $N$ , should be 30-40,
- iii) the number of repetitions of a solution before invoking the back-tracking mechanism should be 5-15,
- iv) the  $k^{\text{th}}$  backward solution selected by the back-tracking mechanism should be equal or close to the number of repetitions of a solution before invoking the back-tracking mechanism,
- v) the adaptive search radius should employ 20-25% of radius reduction, and
- vi) a well educated guess of the search space that is wide enough to cover the global solution is necessary.

The detailed derivation of these settings can be found in [12].

## 4 ATS-Based Design of PID Controller

The ATS-based design of the PID controllers is illustrated in this section. Referring to Fig. 2, the AI tuning block utilizes the ATS method. The parameter tuning process is repeatedly performed to minimize the cost function  $J$  stated in (3)-(4) until the termination criterion is met. In this work, the maximum search round ( $count_{max}$ ) of the search process is the termination criterion. In each search round, 40 neighborhood members are randomly generated, and  $count_{max}$  is set as 50. The back-tracking mechanism will be activated when the current solution cannot be updated for  $Re_{max} = 5$ , where  $Re_{max}$  is the maximum cycling allowance. The following heuristic rule is applied for the search radius reduction: If ( $n_{cycling} = 20$ ) or ( $n_{cycling} = 40$ ) or ( $n_{cycling} = 60$ ) or ( $n_{cycling} = 80$ ), then  $R = 0.8 * R$ , where  $n_{cycling}$  is number of solution cycling detected for the adaptive search radius mechanism, and  $R = 10\%$  of the search space

$$\begin{array}{ll} \text{Minimize} & J \\ \text{Subject to} & t_r \leq t_{r-max}, P.O. \leq P.O._{max}, t_s \leq t_{s-max}, E_{ss} \leq E_{ss-max} \end{array} \quad (4)$$

To evaluate the performance of the proposed design method, the ATS-based design of the PID controllers is applied to some selected benchmark systems suggested by Åström, et al. [8]. Those systems are difficult to design the PID controllers by conventional design methods or tuning rules. The performance specifications are given in Eq. (4), where  $t_r$  is rise time,  $P.O.$  is percent overshoot,  $t_s$  is settling time, and  $E_{ss}$  is steady state error. They are set as the inequality constraints of the optimization problem. The numeric values of these design specifications will be given correspondingly to the dedicated benchmark systems.

Referring to Table 1, nine types of plants are listed in the table with their corresponding design specifications. These plants are fourth order system (entry 1), multiple poles (entry 2), right half zero (entry 3), time delay and lag (entry 4), time delay and double lag (entry 5), fast and slow modes (entry 6), conditionally stable system (entry 7), oscillatory system (entry 8), and unstable system (entry 9), respectively. When the ATS is applied, the ranges of the parameters to be searched in the search spaces must be given. Table 2 gives the ranges of the controller parameters to be searched.

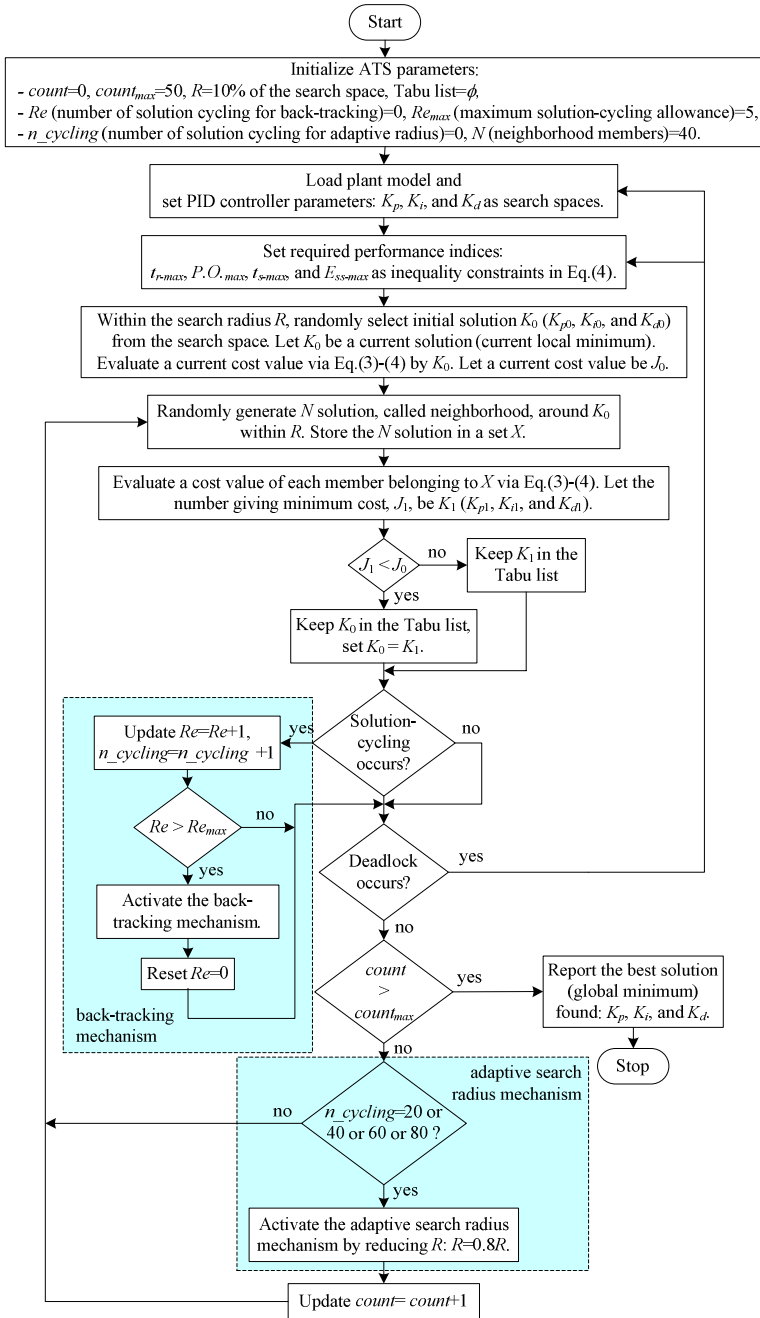


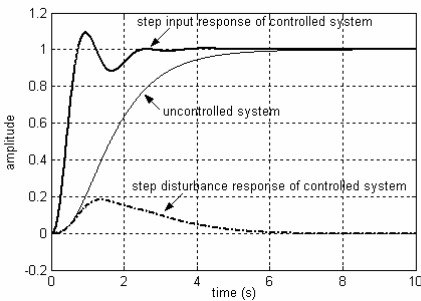
Fig. 3. Diagram of the ATS's implementation for designing PID controllers



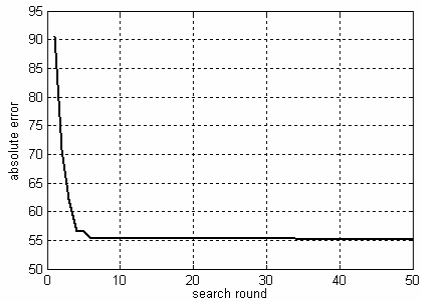
**Table 1.** List of plant models and design specifications

Entry	Plant Models	Design Specifications			
		$t_{r-max}$ (sec)	$P.O._{max}$ (%)	$t_{s-max}$ (sec)	$E_{ss-max}$
1.	$G_{p1}(s) = \frac{1}{(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}, \alpha = 0.5$	1.00	10.00	2.50	0.00
2.	$G_{p2}(s) = \frac{1}{(1+s)^4}$	3.00	15.00	15.00	0.00
3.	$G_{p3}(s) = \frac{1-\alpha s}{(1+s)^3}, \alpha = 0.5$	2.50	10.00	8.00	0.00
4.	$G_{p4}(s) = \frac{1}{(1+Ts)} e^{-s}, T = 10$	3.00	10.00	7.50	0.00
5.	$G_{p5}(s) = \frac{1}{(1+Ts)^2} e^{-s}, T = 10$	5.00	15.00	25.00	0.00
6.	$G_{p6}(s) = \frac{100}{(s+10)^2} \left( \frac{1}{s+1} + \frac{0.5}{s+0.05} \right)$	0.25	10.00	1.50	0.00
7.	$G_{p7}(s) = \frac{(s+6)^2}{s(s+1)^2(s+36)}$	0.30	25.00	2.50	0.00
8.	$G_{p8}(s) = \frac{\omega_0^2}{(s+1)(s^2+2\zeta\omega_0s+\omega_0^2)}, \omega_0 = 1, \zeta = 0.1$	2.50	25.00	10.00	0.00
9.	$G_{p9}(s) = \frac{1}{s^2-1}$	0.10	15.00	0.50	0.00

**Note:**  $t_{r-max}$  is maximum rise time allowance,  $P.O._{max}$  is maximum percent overshoot allowance,  $t_{s-max}$  is maximum settling time allowance, and  $E_{ss-max}$  is maximum steady state error allowance.



**Fig. 4.** Step responses of  $G_{p1}(s)$



**Fig. 5.** Convergence of  $J$  of  $G_{p1}(s)$

The diagram in Fig. 3 reveals the search process of the ATS for designing PID controllers. In this work, the ATS was coded in MATLAB™ for running on a Pentium 4, 1.6 GHz, 256 Mbytes RAM, 40 Gbytes HD. The flow diagram gives a clear view of the

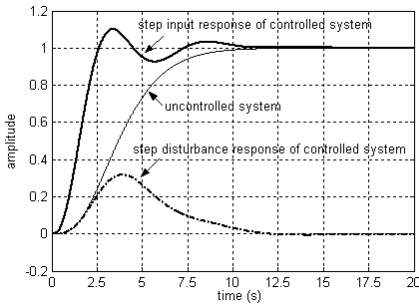


Fig. 6. Step responses of  $G_{p2}(s)$

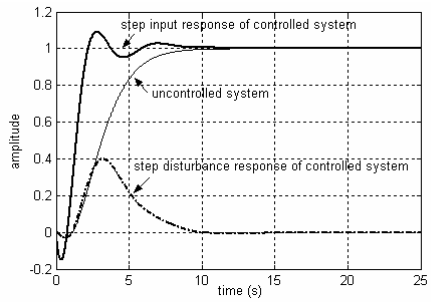


Fig. 7. Step responses of  $G_{p3}(s)$

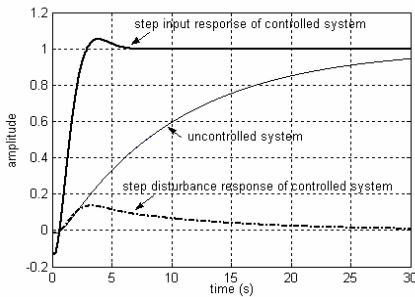


Fig. 8. Step responses of  $G_{p4}(s)$

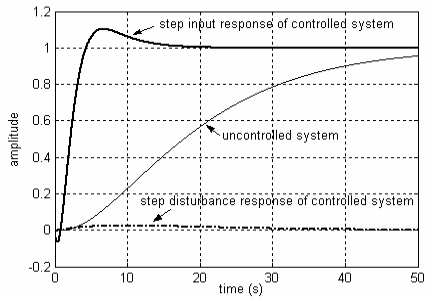


Fig. 9. Step responses of  $G_{p5}(s)$

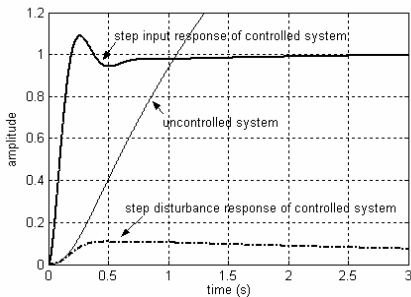


Fig. 10. Step responses of  $G_{p6}(s)$

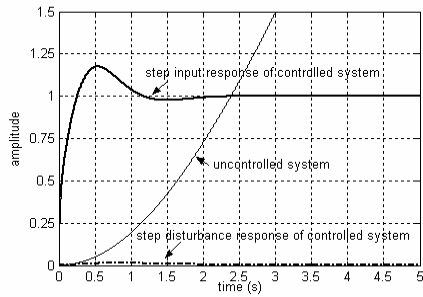


Fig. 11. Step responses of  $G_{p7}(s)$

proposed method for the readers to follow. For more details of the back-tracking and the adaptive search radius mechanisms, the readers should consult the references [12],[13].

As a result for the entry-1 plant, the step input and the step disturbance responses are shown in Fig. 4. Fig. 5 shows the convergence of the cost function  $J$ , as an example. The response curves in Fig. 4 confirm that the resulted controller governs the plant to produce high quality tracking performance as well as rapidly recover from the external disturbance. The controller parameters and the responses are summarizes in Table 3. For the plants of the other entries, the corresponding step input and step

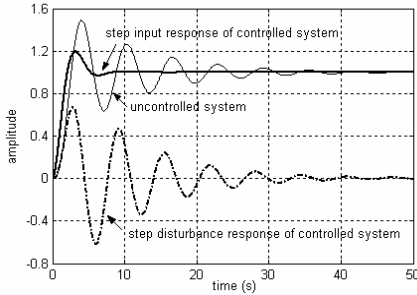


Fig. 12. Step responses of  $G_{p8}(s)$

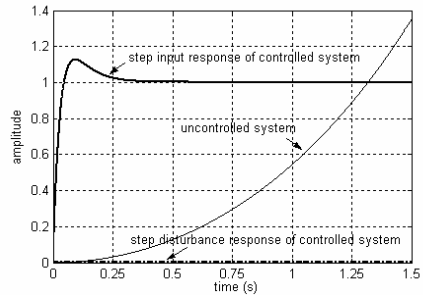


Fig. 13. Step responses of  $G_{p9}(s)$

Table 2. List of Search spaces for ATS-based PID controller design

Entry	Search Spaces			Entry	Search Spaces		
	$K_p$	$K_i$	$K_d$		$K_p$	$K_i$	$K_d$
1.	[0, 10]	[0, 10]	[0, 10]	6.	[0, 10]	[0, 10]	[0, 10]
2.	[0, 5]	[0, 5]	[0, 5]	7.	[0, 80]	[0, 80]	[0, 30]
3.	[0, 5]	[0, 5]	[0, 5]	8.	[0, 5]	[0, 5]	[0, 5]
4.	[0, 10]	[0, 5]	[0, 5]	9.	[400, 500]	[0, 50]	[0, 50]
5.	[0, 20]	[0, 5]	[0, 50]				

Table 3. List of PID controllers and step responses obtained from the ATS method

Entry	PID Controllers			Step Input Responses				Step Disturbance Responses	
	$K_p$	$K_i$	$K_d$	$t_r$ (sec)	$P.O.$ (%)	$t_s$ (sec)	$E_{ss}$	$t_{re}$ (sec)	$P.O.$ (%)
1.	3.78	2.16	2.32	0.80	8.50	2.20	0.00	6.52	19.24
2.	2.17	0.70	2.86	2.54	9.82	11.88	0.00	11.94	32.25
3.	1.80	0.72	1.75	2.40	8.50	7.80	0.00	10.02	40.00
4.	6.62	0.66	0.90	2.65	5.52	6.86	0.00	28.27	13.83
5.	10.78	0.61	39.83	4.25	10.05	17.28	0.00	23.48	2.42
6.	8.33	1.50	1.20	0.22	9.00	1.25	0.00	31.50	10.82
7.	59.43	45.48	24.96	0.25	19.00	2.25	0.00	1.82	1.36
8.	0.24	1.21	1.22	2.20	19.50	7.50	0.00	42.45	68.43
9.	449.86	45.75	44.83	0.08	12.50	0.38	0.00	0.35	0.22

Note:  $t_r$  is rise time,  $P.O.$  is percent overshoot,  $t_s$  is settling time,  $E_{ss}$  is steady state error, and  $t_{re}$  is recovering time.

disturbance responses are illustrated in the Fig. 6 to Fig. 13, respectively. The search convergence curves of these cases are omitted because they have a similar form to that of the entry-1 plant shown in Fig. 5. The readers can notice from Fig. 6 to Fig. 13 that the resulted controllers from the ATS well perform both tracking and regulating objectives with an exception for the oscillatory plant (entry 8). For the entry-8 plant, it will need another dedicated controller to meet the disturbance rejection requirement. Table 3 summarizes the controller parameters and the responses.

## 5 Conclusions

Meeting the stringent requirements of control specifications is achieved by the adaptive tabu search (ATS) as described in this paper. The ATS is flexible and suitable for a variety of optimization problems in which control application is one of the domains. The paper presents the results of obtaining optimum PID controllers for 9 benchmark systems suggested by Åström, et al. [8]. The ATS yields very good results except for the case of step disturbance rejection in the oscillatory plant.

**Acknowledgements.** The authors wish to thank South-East-Asia University and Suranaree University of Technology, Thailand, for the financial supports.

## References

1. Bennett, S.: Development of the PID controller. *IEEE Control System Magazine* (1994) 58-65
2. Kuo, B.J.: *Automatic Control Systems*. 8<sup>th</sup> edn. John Wiley & Sons (2003)
3. Ogata, K.: *Modern Control Engineering*. 4<sup>th</sup> edn. Prentice-Hall (2002)
4. Dorf, R.C.: *Modern Control Systems*. 10<sup>th</sup> edn. New Jersey Prentice-Hall (2005)
5. Ziegler, J.G., Nichols, N.B.: Optimum Settings for Automatic Controllers. *Trans. ASME*, Vol. 64 (1942) 759-768
6. Cohen, G.H., Coon, G.A.: Theoretical Consideration of Retarded Control. *Trans. ASME*, Vol. 75 (1953) 827-834
7. Dwyer, A.O.: *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press (2003)
8. Åström, K.J., Hägglund, T.: Benchmark Systems for PID Control. *IFAC Digital Control: Past, Present and Future of PID Control*. Terrassa. Spain (2000) 165-166
9. Mehrdad, S., Greg, C.: An Adaptive PID Controller based on Genetic Algorithm Processor. *Proc. IEE Conf. on Genetic Algorithm in Engineering System* (1995) 88-93
10. Mitsukura, Y., Yamamoto, T., Kaneda, M.: A Design of Self-tuning PID Controller using a Genetic Algorithm. *Proc. American Control Conference* (1999) 1361-1365
11. Whidborne, J.F.: A Genetic Algorithm Approach to Design Finite-Precision PID Controller Structures. *Proc. American Control Conference* (1999) 4338-4342
12. Puangdownreong, D., Kulworawanichpong, T., Sujitjorn, S.: Finite Convergence and Performance Evaluation of Adaptive Tabu Search. *Lecture Notes in Artificial Intelligence*, Vol. 3215. Springer-Verlag, Berlin Heidelberg (2004) 710-717
13. Puangdownreong, D., Areerak, K-N., Srikaew, A., Sujitjorn, S., Totarong, P.: System Identification via Adaptive Tabu Search. *Proc. IEEE Int. Conf. on Industrial Technology (ICIT'02)*, Vol. 2 (2002) 915-920

# Author Index

- Acevedo, Javier II-238  
Adams, Rod I-822  
Ahmed, Bilal II-300  
Al-Jumeily, Dhiya II-123  
Alimi, Adel M. I-240  
Allahverdi, Novruz II-467  
Allende, Héctor II-355, II-554  
Altun, Adem Alpaslan II-467  
Álvarez, José L. I-39  
Amari, Shun-ichi II-271  
Andrejková, Gabriela I-404  
Anisetti, Marco I-684  
Arslan, Ahmet I-694
- Bae, Hyeon-Deok II-534  
Baer, Philipp A. I-167  
Bağ, Michał II-133  
Barreira, Noelia I-202  
Barrón, Ricardo II-55  
Becerra, Carlos II-554  
Beliczynski, Bartłomiej II-46  
Bellandi, Valerio I-684  
Ben Amor, Heni II-641  
Biçici, Ergun I-739  
Bielecki, Andrzej II-133  
Bielski, Conrad II-500  
Bilgin, Mehmet Zeki II-713  
Borkowski, Adam I-102  
Bravo, Crescencio I-649  
Browne, Jim I-498  
Byun, Yeun-Sub I-657
- Cai, Nian II-582  
Chacón, Máx II-355  
Chai, Zhilei I-376, I-394  
Chakraborty, Uday K. I-77  
Chandran, Harish I-11  
Chang, Ping-Teng I-631  
Charuwat, Thitipong I-230  
Chen, Ping-Jie II-246  
Chen, Ta-Cheng I-526  
Cheng, Jao-Hong I-614  
Chiu, Deng-Yiv II-246  
Chiu, Singa Wang II-525  
Chiu, Yuh-Wen I-614
- Cho, Dal-ho II-440  
Cho, Seong Jin II-300  
Choi, Dae-Young I-517  
Choi, Hun II-534  
Choi, Jae-Seung II-153  
Choi, Jeoung-Nae I-622  
Choi, Se-Hyu I-306  
Choi, Seungjin II-271  
Choi, YoungSik I-588  
Chong, Kil To II-676  
Chongchao, Huang I-341  
Choraś, Michał II-407, II-424  
Choraś, Ryszard S. II-407  
Chou, Chih-Hsun I-604  
Chung, Chung-Yu II-525  
Chung, Gyo-Bum II-738  
Chung, Yongwha II-432  
Cichocki, Andrzej II-271, II-373  
Cieslik, Dominik II-624  
Couchet, Jorge I-730  
Cutello, Vincenzo I-93  
Cyganek, Bogusław II-508  
Czajewski, Witold II-633
- Dąbrowski, Paweł I-194  
Davey, Neil I-822  
Deguchi, Toshinori II-37  
Deißler, Tobias II-616  
Dereli, Türkay I-508  
Dioşan, Laura II-218  
Dobnikar, Andrej II-63  
Dominik, Andrzej I-772  
Dorado, Julián I-276  
Dreżewski, Rafał I-67  
Du, Wei I-296  
Duque, Rafael I-649  
Dvořák, Jakub I-721  
Dzemyda, Gintautas II-179, II-544  
Dzielinski, Andrzej I-414
- Ebert, Alfons II-492, II-616  
Erig Lima, Carlos R. I-159
- Fabijański, Paweł I-640  
Faling, Gui I-341

- Fang, Wei I-376  
 Ferreira, Enrique I-730  
 Folan, Paul I-498  
 Fonseca, André I-730  
 Fraser, Robert I-758
- Gabrijel, Ivan II-63  
 Galán-Marín, Gloria I-461, II-98  
 Galbiati, Jorge II-554  
 Gambin, Anna I-422  
 Gámez, José A. I-806  
 Gammoudi, Jamil I-148  
 García, Cristina II-667  
 Gegúndez, Manuel E. I-39  
 Geihs, Kurt I-167  
 Ghazali, Rozaida II-123  
 Gil, Pedro II-238  
 Glasgow, Janice I-758  
 Godoy Jr., Walter I-358  
 Gómez-Tato, Andrés II-208  
 González, Jesus I-85  
 González-Castaño, Francisco J. II-208  
 González de-la-Rosa, Juan-José I-782  
 Górriz, Juan-Manuel I-782  
 Grześ, Marek I-1  
 Guillén, Alberto I-85  
 Güneş, Salih II-338  
 Guo, Xinchun II-189  
 Güven, Aysegül II-338  
 Gwun, Ou-Bong II-590
- Hahn, Minsoo II-382  
 Hamdani, Tarek M. I-240  
 Han, Changwook I-257  
 Han, Liyan I-314  
 Han, Song-yi II-440  
 Han, Soowhan I-257  
 Hembecker, Fernanda I-358  
 Herrera, Luis J. I-85  
 Hlaváčková-Schindler, Kateřina I-790  
 Hsieh, Yi-Chih I-526  
 Huang, Hai I-314  
 Hussain, Abir Jaafar II-123  
 Hwang, Hyung-Soo I-622
- Ibáñez, Óscar I-202  
 Ikemoto, Shuhei II-641  
 Im, Dae-Yeong II-730  
 İnal, Melih I-266  
 Ishiguro, Hiroshi II-641
- Ishii, Naohiro II-37  
 Ivanikovas, Sergejus II-179  
 Iwanowski, Marcin II-606
- Janežič, Dušanka II-399  
 Jang, Seong-Whan I-666  
 Jarur, Mary Carmen II-107  
 Jedruch, Wojciech I-386  
 Jedrzejowicz, Piotr I-480  
 Jeon, Gwanggil I-684  
 Jeong, Dae Sik II-415  
 Jeong, Jechang I-684  
 Jeong, Jong-Cheol II-364  
 Jeong, SangBae II-382  
 Jiang, Jingqing I-562  
 Jo, Geun-Sik II-71  
 Joseph, Shaine I-77  
 Jung, Bernhard II-641  
 Jung, Jin-Guk II-71  
 Jung, Seunghwan II-432
- Kacalak, Wojciech I-596  
 Kaczorek, Tadeusz II-694  
 Kainen, Paul C. II-11  
 Kang, Byung Jun II-415  
 Kang, Hyung W. I-77  
 Kara, Sadık II-338  
 Karci, Ali I-450  
 Karray, Fakhri I-240  
 Kawaguchi, Masashi II-37  
 Khan, Mohammed A.U. II-300  
 Kim, Eun-Mi II-364  
 Kim, Gwang-Ha II-290  
 Kim, Hyongsuk II-346  
 Kim, Hyun-Ki I-666  
 Kim, Il-hwan II-722  
 Kim, Jinhwa I-830  
 Kim, Kwang-Baek II-290, II-572  
 Kim, Mi-Young I-814  
 Kim, Min-Soo I-657  
 Kim, Minhwan II-572  
 Kim, Sang-Chul II-659  
 Kim, Sungshin II-290  
 Kim, Tae-Seong II-300  
 Kim, Young-Chul II-676  
 Kisiel-Dorohinicki, Marek I-138  
 Kleiber, Michał I-570  
 Koh, Eun Jin II-517  
 Kokosiński, Zbigniew I-211  
 Konc, Janez II-399

- Kong, Jun II-309  
 Koperwas, Jakub I-702  
 Köppen, Mario I-323  
 Korbicz, Józef II-19  
 Kozak, Karol II-327  
 Kozak, Marta II-327  
 Kozłowski, Bartosz I-49  
 Krasnogor, Natalio I-93  
 Krętoski, Marek I-1  
 Kulworawanichpong, Thanatchai I-230  
 Kumeresh, Arjun I-11  
 Kurasova, Olga II-544  
 Kůrková, Věra II-11  
 Kupis, Paweł I-23  
 Kusiak, Magdalena I-432  
 Kwarciany, Krzysztof I-211  
 Kwolek, Bogdan II-599
- Lafuente, Sergio II-238  
 Lagoda, Ryszard I-640  
 Lai, Choi-Hong I-394  
 Lai, Kin Keung II-262  
 Lawryńczuk, Maciej II-143  
 Lee, Bae-Ho II-364  
 Lee, Chong Ho I-286  
 Lee, Dae-Young II-534  
 Lee, Eui Chul II-415  
 Lee, Imgeun I-257  
 Lee, In-Tae I-666  
 Lee, Inbok I-554  
 Lee, Jae-kang II-722  
 Lee, Jee-hyong I-441  
 Lee, Ji-Yeoun II-382  
 Lee, Ju-Sang II-730  
 Lee, Keon-myung I-441  
 Lee, Kidong I-830  
 Lee, Sangyoung II-440  
 Lee, Sungju II-432  
 Lee, Sungyoung II-300  
 Lee, Yung-Cheng I-526  
 Lengeňová, Helena I-404  
 Lévano, Marcos II-355  
 Li, Hongzhi II-309  
 Li, Ming II-582  
 Li, Qing I-498  
 Liang, Yanchun I-296, I-562  
 Liberski, Paweł II-346  
 Lin, Hong-Dar II-525  
 Lin, Kuo-Ping I-631  
 Lipiński, Dariusz I-596
- Lipinski, Piotr II-391  
 Liu, Fan-Yong I-674  
 Lloret, Isidro I-782  
 Lopatka, Rafal I-414  
 Lopes, Heitor S. I-159, I-358  
 López-Rodríguez, Domingo I-461, II-98  
 Lotfi, Naser I-110  
 Lotfi, Shahriar I-110  
 Lotrič, Uroš II-254  
 Lu, Yinghua II-309  
 Luo, Zhi-Jie I-604
- Maddouri, Mondher I-148  
 Maldonado, Saturnino II-238  
 Mańdziuk, Jacek I-23, I-432  
 Manrique, Daniel I-730  
 Mariańska, Bożena II-318  
 Markiewicz, Tomasz II-318  
 Martínez-Álvarez, Rafael P. II-208  
 Masouris, Michael II-457  
 Mateo, Juan L. I-806  
 Mati, Michal I-404  
 Medvedev, Viktor II-179  
 Merabti, Madjid II-123  
 Mérida-Casermeiro, Enrique I-461,  
 II-98  
 Middelmann, Wolfgang II-492, II-616  
 Minato, Takashi II-641  
 Mohamed Ben Ali, Yamina I-128  
 Moon, Daesung II-432  
 Mora, Marco II-107  
 Moreno, José Alí II-667  
 Mroczkowski, Piotr II-424  
 Mrugalski, Marcin II-19  
 Muñoz, Antonio Moreno I-782
- Nam, Mi Young II-517  
 Nicosia, Giuseppe I-93  
 Nikodem, Piotr I-102  
 Noh, JiSung I-588  
 Nowak, Hans II-355
- Ogiela, Marek R. II-477  
 Ogryczak, Włodzimierz I-578  
 Oh, Sung-Kwun I-622, I-666  
 Oh, Tae-seok II-722  
 Oltean, Mihai I-220, II-218  
 Orłowska, Maria I-536  
 Ortiz-de-Lazcano-Lobato, Juan M.  
 I-461, II-98

- Osowski, Stanisław II-318, II-373  
 Ospina, Juan I-120  
 Özcan, Ender I-366  
  
 Pae, Ho-Young II-364  
 Paechter, Ben I-85  
 Pai, Ping-Feng I-631  
 Palacios, Pablo I-39  
 Pang, Yunjie I-546  
 Park, Doo-kyung I-441  
 Park, Hyun-Ae II-440  
 Park, Jang-Hyun II-730  
 Park, Jong-Ho II-676  
 Park, Junghee I-830  
 Park, Kang Ryoung II-415, II-440  
 Park, Kyo-hyun I-441  
 Park, Seung-Jin II-153  
 Parsa, Saeed I-110  
 Pavone, Mario I-93  
 Pazos, Alejandro I-276  
 Pearson, David W. I-767  
 Pecuchet, Jean-Pierre II-218  
 Pempera, Jaroslaw I-194  
 Penedo, Manuel G. I-202  
 Pettersson, Frank II-115  
 Piao, Chang Hao I-286  
 Pierluissi, Luis I-31  
 Pinto, Pedro C. I-350  
 Plemmons, Robert II-271  
 Podolak, Igor T. I-749  
 Polat, Kemal II-338  
 Polat, Övünç II-161  
 Pomares, Hector I-85  
 Prodan, Lucian I-174  
 Puangdownreong, Deacha II-747  
 Puerta, José M. I-806  
 Puntonet, Carlos G. I-782  
  
 Qi, Miao II-309  
  
 Rabuñal, Juan I-276  
 Rasheed, Tahir II-300  
 Ratajczak-Ropel, Ewa I-480  
 Raudys, Sarunas I-711, II-1  
 Rhee, Phill Kyu II-517  
 Ribeiro, Bernardete II-199, II-228  
 Rivero, Daniel I-276  
 Robinson, Mark I-822  
 Rocco S., Claudio M. I-31  
 Rodríguez-Hernández, Pedro S. II-208  
  
 Rogozan, Alexandrina II-218  
 Rojas, Ignacio I-85  
 Romaszkiwicz, Andrzej I-578  
 Rosas, Lorna V. II-564  
 Rudnicki, Marek II-281  
 Ruican, Cristian I-174  
 Runkler, Thomas A. I-350  
 Rust, Alistair G. I-822  
 Ryoo, Young-Jae II-685, II-730  
 Rysz, Andrzej II-373  
  
 Sadiq, Shazia I-536  
 Şakiroğlu, Ayşe Merve I-694  
 Sandou, Guillaume I-332  
 Sanguineti, Marcello II-11  
 Santos, José I-202  
 Savický, Petr I-721  
 Savvopoulos, Anastasios I-837  
 Saxén, Henrik II-115  
 Sbarbaro, Daniel II-107  
 Sena Daş, Gülesin I-508  
 Serban, Ana-Talida I-332  
 Sharabi, Offer I-822  
 Shin, Yun-su II-722  
 Shukla, Pradyumn Kumar I-58  
 Siegmann, Philip II-238  
 Sienkiewicz, Rafal I-386  
 Silva, Catarina II-228  
 Silva, Rafael R. I-159  
 Siwik, Leszek I-67, I-138  
 Skoneczny, Slawomir II-624  
 Smith, Darren I-488  
 Smutnicki, Czeslaw I-194  
 Sohn, Sang-Wook II-534  
 Soille, Pierre II-500, II-606  
 Son, Eun-Ho II-676  
 Song, Ha Yoon I-554  
 Song, Ju-Whan II-590  
 Sossa, Humberto II-55  
 Sotiropoulos, Dionisos N. I-837  
 Sousa, João M.C. I-350  
 Staniak, Maciej II-633  
 Stapor, Katarzyna II-327  
 Stasiak, Bartłomiej II-27  
 Stathopoulou, Ioanna-Ourania II-449  
 Šter, Branko II-63  
 Strumiłło, Paweł II-281  
 Strzelecki, Michal II-346  
 Suh, Jae-Won II-534  
 Sujitjorn, Sarawut II-747



- Sun, Fangxun I-296  
 Sun, Jun I-376, I-394  
 Sun, Yi I-822  
 Sung, Andrew H. II-228  
 Świdorski, Bartosz II-373  
 Szatkiewicz, Tomasz II-80  
 Szczurek, Ewa I-422
- Tadeusiewicz, Ryszard II-477  
 Tambouratzis, Tatiana II-169,  
 II-457, II-649  
 te Boekhorst, Rene I-822  
 Thoennesen, Ulrich II-492, II-616  
 Tian, Lei I-314  
 Tiesong, Hu I-341  
 Trebar, Mira II-254  
 Trojanowski, Krzysztof I-184  
 Tsihrintzis, George A. II-449, I-837
- Uddin, Mohammed Nazim II-71  
 Udrescu, Mihai I-174  
 Unold, Olgierd I-798
- Vargas, Héctor II-564  
 Vázquez, Roberto A. II-55  
 Vejmelka, Martin I-790  
 Vélez, Mario I-120  
 Venkateswaran, Nagarajan I-11  
 Viet, Nguyen Hoang I-570  
 Virvou, Maria I-837  
 Vladutiu, Mircea I-174
- Walczak, Krzysztof I-702  
 Walczak, Zbigniew I-772  
 Wałędzik, Karol I-432  
 Waller, Matias II-115  
 Wang, Chaoyong II-189  
 Wang, Hui II-582  
 Wang, Jin I-286  
 Wang, Qing I-498  
 Wang, Rujuan II-309  
 Wang, Shouyang II-262  
 Wang, Shuqin I-296  
 Wang, Xin I-546  
 Wang, Xiumei I-296
- Wang, Yan I-296  
 Wang, Yijing II-88, II-704  
 Wang, Yunxiao I-546  
 Wang, Zhengxuan I-546  
 Wawrzyński, Paweł I-470  
 Webb, Barbara I-488  
 Weise, Thomas I-167  
 Wessnitzer, Jan I-488  
 Wojciechowski, Jacek I-772  
 Won, Jin-Myung I-240  
 Woo, Young Woon II-572  
 Wu, Chunguo I-562, II-189  
 Wysocka-Schillak, Felicja I-248
- Xianing, Wu I-341  
 Xiao, TianYuan I-498  
 Xu, Wenbo I-376, I-394
- Yan, Jiang I-341  
 Yang, Hyong-Yeol II-730  
 Yang, Jie II-582  
 Yang, Jinhui I-562  
 Yang, Li-An I-604  
 Yatsymirskyy, Mykhaylo II-27, II-391  
 Yeh, Chung-Hsing I-614  
 Yıldırım, Tülay II-161  
 Yılmaz, Murat I-366  
 Yoon, Tae-bok I-441  
 Yoon, Tae-jun II-722  
 Yoshida, Kaori I-323  
 Yu, Kun-Ming I-604  
 Yu, Lean II-262  
 Yu, Xiao I-546  
 Yuan, Bo I-536  
 Yuret, Deniz I-739
- Zalewska, Anna II-346  
 Zanella, Vittorio II-564  
 Zdunek, Rafal II-271  
 Zhang, Na I-562  
 Zhang, Wenbo II-189  
 Zhou, Chunguang I-296  
 Zhou, Jian I-498  
 Zhou, Jiayi I-604  
 Zuo, Zhiqiang II-88, II-704